Dynamic Analysis of Biped Robot Locomotion and Design, Development, Experimentation of a Statically Stable Biped Robot

A Thesis Submitted

In Partial Fulfillment of the Requirements

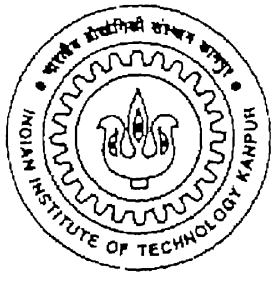
For the Degree of

Master of Technology

by

PRASAD KULKARNI

Y210526



To the

Department of Mechanical Engineering

Indian Institute of Technology Kanpur

India

May, 2004

CERTIFICATE

It is certified that the work contained in this thesis entitled "***Dynamic Analysis of Biped Robot Locomotion and Design, Development, Experimentation of a Statically Stable Biped Robot,***" by Mr. Prasad Kulkarni (roll no. 210526), has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. Ashish Dutta
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, 208016

May, 2004

Dedicated to

My Parents and My Brother

Acknowledgement

With immense pleasure I express my sincere gratitude, regards and thanks to my supervisor Prof. Ashish Dutta for his excellent guidance, invaluable suggestions and continuous encouragement at all the stages of my research work. His interest and confidence in me was the reason for all the success I have made. I have been fortunate to have him as my guide as he has been a great influence on me, both as a person and as a professional.

I am extremely thankful to Prof. B. Dasgupta for his invaluable suggestions and generous help during the entire period of this work. I could approach him any time without hesitation.

I am thankful to Prof. A. Mukarjee for extending the facilities of the Center for Robotics for the completion of the work. It was a pleasure to be associated with Center for Robotics and would like to thank all the staff members and my colleagues Hari, Madan, Mooshir, Pankaj, Sandeep, Shahzad, Srikant, Srinivas and Surya for helping me in all stages of my work and making the Center for Robotics a great place to work in.

I would like to thank my friends Anoop, Anupam, Bijoy, Deepak, Kisun, Prince, Pritam, Santanu, Soumya, Sudhish and all my classmates for their smiles and friendship making the life at I.I.T. Kanpur enjoyable and memorable.

Above all, I am blessed with such caring parents. I extend my deepest gratitude to my parents and my elder brother for their invaluable love, affection, encouragement and support.

Prasad Kulkarni

I.I.T. Kanpur

Abstract

Biped robots have better mobility than conventional wheeled robots as they can move over obstacles, climb stairs etc. However they have to balance themselves by positioning their links which makes their control very difficult. In this thesis a stable walking gait for a dynamically stable biped robot having 8 DOF, based on ZMP approach has been analyzed. A series of stable configurations have been computed following which the biped can have dynamic stability during gait. In this method of gait generation, the trajectory of the ankle of the foot in air is assumed for a given step length and then a smooth hip trajectory is calculated iteratively such that the ZMP is inside the supporting foot perimeter. The results of the simulation prove that dynamic stability during biped gait can be obtained by controlling the ZMP.

As most humanoid robots developed so far are extremely complex, because it is required to control a large number of actuators, this thesis proposes a simple biped humanoid robot having only four actuators. This statically stable biped robot can follow desired trajectories, climb stairs, walk over inclinations and emulate simple human actions like dancing. The biped has been designed, developed and fabricated. The proposed robot was first tested in simulations for trajectory following, obstacle avoidance, stair climbing and then experiments were carried out. The results of the experiments prove the usefulness of the proposed design.

Contents

Chapter 1

Introduction

This thesis is about understanding active dynamic balancing of biped robots and the development of a simple walking mechanism that is not biologically inspired but is adequate for simple locomotion. The first question which needs to be answered is why do we need to study legged locomotion at all? There are many reasons why we should. The philosophical answer is that it will enable us to understand more about human and animal locomotion. The more practical answer is that legs have advantage over wheel in applications like travel over difficult terrain, obstacle avoidance, stair climbing etc. In future humanoid robots would be living in the homes of people and in such cases robots would need to climb stair, avoid obstacles in an environment designed mainly for human locomotion. Legged systems also have an active suspension that decouples the path of the legs from the path of the body. The advantages of legs are also borne by evolution, as biological systems (humans/animals) have legs and not wheel for locomotion. This is because humans use their legs for walking, running, climbing stairs, sitting down etc. and a wheel could never satisfy all these requirements. However the reason as to why legged robots we still in an incipient stage is because despite excellence in using our own legs, we are still at a primitive stage in understanding the control principles that underlie walking and running. An example which illustrates how biological systems control the natural dynamics of a system is to see a child riding a bicycle. After a few attempts the child learns to control the natural dynamics involved. However it is still virtually impossible for a robot to learn how to ride a bicycle.

The main objective of the thesis may be stated as:

a) Analysis of active dynamic locomotion with an emphasis on understanding balancing and control.
b) Development of an alternate walking robot having only four actuators, a statically stable biped robot which can follow trajectories, climb stairs etc.
c) Experimental evaluation of the developed statically stable biped robot.

Dynamic walking mechanisms balance themselves actively as they travel. Unlike in a static robotic arm, the geometry and configuration of the system alone does not provide an adequate model of behavior. Chapter 3 focuses on the aspects of dynamic biped locomotion with emphasis on balance. The term 'balance' here means that the biped can stand on one leg or two legs without toppling over. In order to analyze the system the dynamic system is converted into a static system by applying 'D'Alemberts' principle. It is then analyzed using the laws of mechanics to ensure that there is no unbalanced force or moment acting at the contact point on the foot. This point is referred to as the Zero Moment Point (ZMP) and it always has to be inside the foot perimeter. The objective of ZMP control ensures that for following a given trajectory by the foot the configuration of all the links are such that the ZMP is inside the foot perimeter on the ground. The exact procedure for deriving the configuration of all the links is defined. A few simulations illustrate that the robot actually walks by balancing itself.

Most humanoid robots built by researchers were inspired by biological locomotion, in that biped robots have joints at the knees, ankle and hips. This has resulted in humanoid robots having ten or more degrees of freedom, similar to human legs. Controlling such a large number of actuators is extremely complicated. As a result of laser like focus on human like legs, other methods of locomotion which might be simple have not been sufficiently studied. One such example has been developed which is not human like in structure but uses only four actuators and can follow given trajectories, climb stairs etc. Chapter 4 discusses about the development of a statically stable biped robot. The basic mechanical design of the robot and actuation of the legs are discussed. The robot can follow desired trajectories and also avoid obstacles, which form the main focus of Chapter 5. Two methods, roadmap method and potential field method are explained which were used to find a path in a workspace containing obstacles. The robot can also climb stairs and the three procedures that can be used are discussed. In chapter 6 experiments conducted and results are illustrated. Finally in Chapter 7 conclusions are given.

Chapter 2

Literature Survey

Humanoid robotics is a relatively new area of research and a true humanoid was built recently in 2000, by Honda Motors called 'ASIMO' (Advanced step in Innovative MObility). However the idea of biped locomotion has been around for many decades, mainly as a study of human walk (GAIT) in biological sciences and prosthetic applications. Early researchers have studied legged locomotion by simulating, building, and controlling walking, hopping, and running robots. There have been quite many passive walking robots/toys built which completely rely on their natural dynamics and the gravitational force in order to walk. McGeer [1] explored passive walking and showed that a system that has no sensors, actuators, or any sort of a brain can walk downhill, if appropriate hardware geometry is used. One of the advantages of these passive walkers is that they are easy to build and they do not require actuators, sensors, or controllers in order to make them move. However these robots have limited capabilities as they cannot walk up a slope or follow a desired trajectory etc. After McGeer several other researchers like Mariano Garcia [2, 3, 5], Anindya Chatterjee [2, 3, 5, 6], Andy Ruina [2-7], Michel Coleman [3-7], Collins [4] and Wisse [4] extended McGeer's work and made several two dimensional and three dimensional passive walkers. John Camp [8] proposed a powered passive biped, which balances itself by gravity, but for walking it needs some power from the actuators. Sugimoto and Osuka [9] worked on controlling a passive biped.

The focus of many studies has been dynamic walking mechanisms such as hopping robots and dynamic bipeds. Raibert and Brown [10] developed a 2D one-legged hopping machine. Hodgins et al [11] developed a planner running biped working on hopping principle. Zerrugh *et al.* [12] have investigated the walking pattern for a biped robot by first recording human kinematic data. Extending the minimum-energy walking method to flat ground and uphill slopes, Channon *et al.* [13], Rostami *et al.* [14], and Roussel *et al.* [15] have proposed methods of gait generation by minimizing the cost

function of energy consumption. Silva *et al.* [16] have investigated the required actuator power and energy by adjusting walking parameters.

Since a dynamic biped robot tends to tip over easily, it is necessary to take stability into account, when determining a walking pattern. Different researchers have proposed different control strategies for stability. Zheng *et al.* [17] have proposed a method of gait synthesis for static stability. Chevallereau *et al.* [18] have discussed dynamic stability when tracking a low energy reference trajectory. Unfortunately, this low-energy reference trajectory does not necessarily satisfy the stability constraint. To ensure the dynamic stability of a biped robot, Vukobratovic [19] studied the stability of a biped and proposed the concept of Zero Moment Point (ZMP). Takanishi *et al.* [20], Shih *et al.* [21, 22, 23], Hirai *et al.* [24] and Dasgupta *et al.* [25] have developed this method of walking pattern synthesis based on ZMP. Basically, these investigations first design a desired ZMP trajectory, then derive the hip motion or torso motion required to achieve that ZMP trajectory. The advantage of this method is that the stability margin can be large if the desired ZMP is always near the center of the stable region. However, since the change of the ZMP due to hip motion is limited, not all desired ZMP trajectories can be achieved. Furthermore, to achieve a desired ZMP trajectory, the hip acceleration may be very large. In this case, since the torso is relatively massive, energy consumption increases, and control of the task execution of the upper limbs becomes difficult. Therefore, it is desirable to obtain hip motion without first designing the desired ZMP trajectory.

A biped robot capable of walking on various ground conditions, such as level ground, rough terrain, and in obstacle filled environments, must be capable of various types of foot motion. Most previous researches have described foot trajectories generated by polynomial interpolation. As a result when there are various constraints such as ground conditions and various foot motions, the order of the polynomial is very high, its computation is difficult, and the trajectory may oscillate. In order to avoid this problem, Shih [21] presented a method for producing foot trajectories by cubic spline interpolation. Unfortunately, Shih only discussed the implementation of simple boundary constraints and a constant foot angle. The work of Shih was extended by Qiang Huang *et al.* [26],

where they have varied the foot angle but they have assumed the hip trajectory to be known.

Apart from control of ZMP several researchers have used the inverted pendulum approach for controlling dynamic stability. In this approach the biped can be modeled as an inverted pendulum. Dynamic balance can be achieved by constantly monitoring the robot's acceleration and adapting the corresponding leg moments. Ching Long Shih [27] and Kajita and Tani [28] have shown the basic working of a biped in inverted pendulum mode. Caux *et al* [29], Park and Kim [30], Sugihara *et al* [31] and Ohnishi [32] are also some of the researchers who worked on this concept.

Perhaps the most recent approach for controlling a biped is by Hasegawa *et al* [33], in which they have used genetic algorithms for finding out the most energy efficient gait.

The first humanoid robot was built in Waseda University in 1973 called 'WL-5' [34]. Since then several robots have been built which ultimately culminated in the development of the Honda Humanoids [35], P1, P2, P3 and the latest of the series named "ASIMO". These bipeds can walk straight, turn, climb up stairs, dance etc. Recently Sony Company has also launched a Biped Robot "SDR-2".

Very few researchers have studied the statically stable biped robots although several mechanical toys have been made which are statically stable. Ching-Long Shih and Chien-Jung Chiou [36] have developed a statically stable biped "BR-1", having 6 degrees of freedom.

Chapter 3

Analysis of Biped Robot Locomotion

The main objective of this chapter is to analyze dynamically stable biped locomotion. It is well known that the biped will be stable if the ZMP lies inside the perimeter of foot on the ground. A series of stable configurations were calculated based on ZMP, following which the biped can have a balanced walk for a given step length and velocity. In order to calculate the stable configurations, the trajectory of the leg in air is predefined and the hip trajectory is calculated and based on this, using inverse kinematics, all configurations are calculated.

A biped robot uses a periodic gait for walking. In the periodic gait first both the legs are on the ground then one leg is lifted and advanced forward. Next both the legs are again on the ground and then next the rear leg is lifted and moved to the front. Hence the robot is alternately on one foot and then on two feet. On a one foot stance the whole body weight and dynamic forces are supported by the foot on the ground and hence the balancing of the robot is the major criterion. Similar to humans, a biped balances itself by positioning its links such that it does not fall. Several actuators are used at the joints for positioning the links and forwarding the body. Figure 3.1 shows different biped classification based on the number of actuators used and the design of biped,

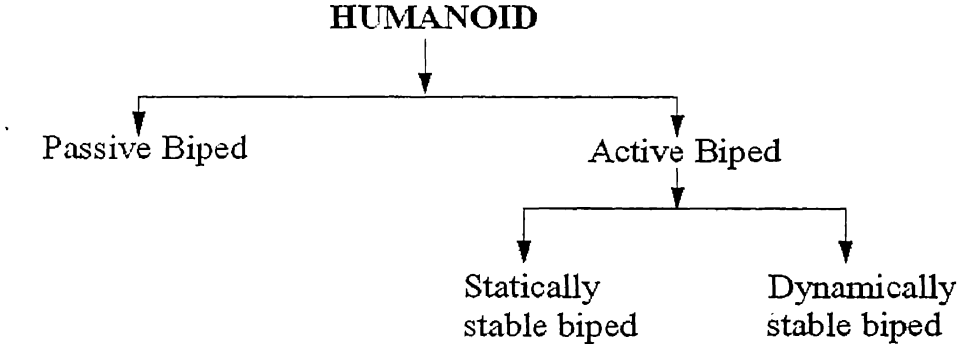


Figure 3.1. Classification of humanoid robots.

3.1 Passive Walker

The passive dynamic biped does not require any actuators. They are walking machines that walk on inclines and they consist of rigid bodies that make collision and rolling contact with the ground surface. They are powered by gravity and natural dynamics and have very little maneuverability. Energy lost in friction and collision is recovered from gravity. These bipeds are dynamic because they are balanced only when they are in motion and they are not stable (can not stand) in static condition. The first model of passive dynamic walker was made by McGeer [1]. Figure 3.2 shows the model proposed by McGeer. A few other passive dynamic biped models have also been developed by several other researchers. Figure 3.3 shows the three dimensional biped developed by Collins et al. [4].

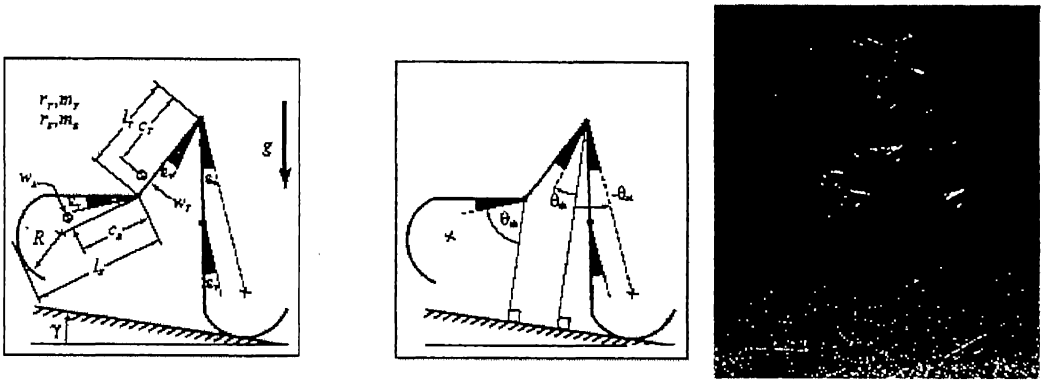


Figure 3.2. McGeer's passive dynamic biped.

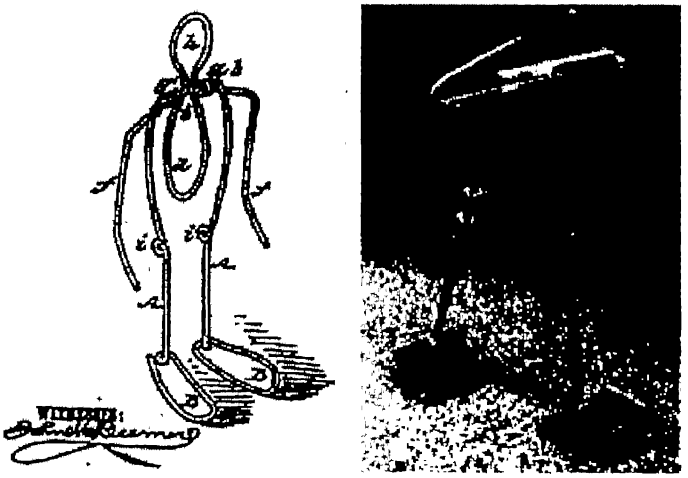


Figure 3.3. Three dimensional passive biped robot.

Figure 3.4 shows the powered passive biped proposed by Camp [8]. In this biped two actuators have been attached to the two ankles. The biped mainly walks passively but for a small interval of time one actuator attached to the leg on the ground is powered so that the other leg is raised up and does not collide with the ground when swinging.

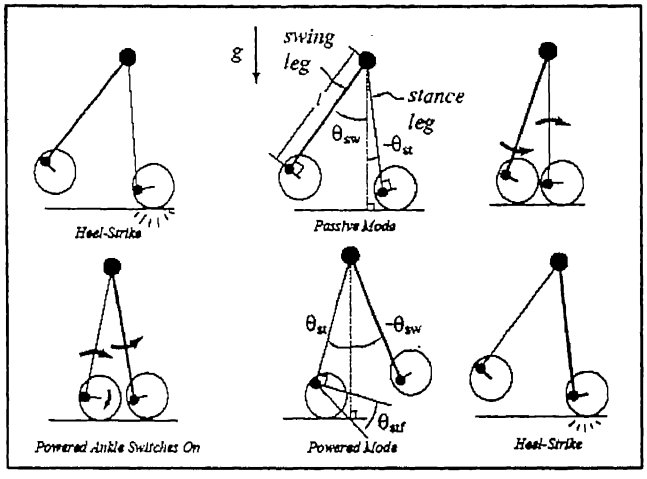


Figure 3.4. Camp's powered passive biped robot.

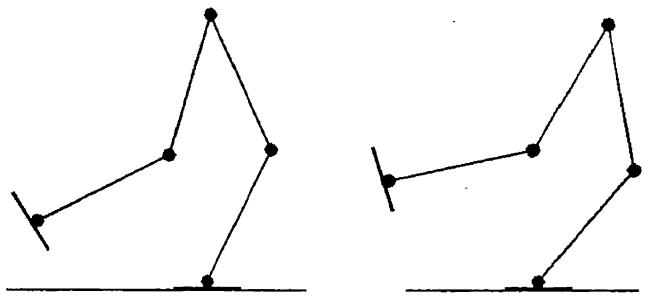
3.2 Statically Stable Biped

A statically stable robot always has the projection of the center of gravity inside the foot contact perimeter (or support region connecting all legs on ground), such that it will always be balanced. They can be of different types such as hexapod, quadruped or biped. In the case of the hexapod and quadruped the center of gravity will always lie inside the support polygon (polygon drawn by connecting the foot of the stance leg on the ground as vertices). Since the biped is always balanced, balancing is not an important criterion in these bipeds. Hence the number of actuators is less than that of the dynamic biped. A statically stable biped robot can walk and follow any desired trajectory by using only three actuators, whereas for the same task a dynamic biped needs at least ten actuators.

3.3 Dynamically stable biped

A Dynamically stable biped robot balances itself by changing the position of its links, such that the ZMP is inside the foot perimeter. Unlike a statically stable biped robot, which is balanced for all configurations, a dynamically stable biped has several

unbalanced configurations as well. Hence for walking and trajectory following, a series of balanced configurations are to be selected. Figure 3.5 shows the balanced and unbalanced configuration of the dynamic biped.



(a) (b)

Figure 3.5. (a) Balanced and (b) unbalanced configurations of the dynamically stable biped.

The ZMP is the point about which sum of all forces and moments of all the masses of the biped is zero. This point must lie inside the supporting foot region. If ZMP point falls out side the supporting foot region then one or more moments would result in a resulting moment, destabilizing the robot. If at least one foot of the robot is on the ground, ZMP approach can be used to find out the stability of the biped.

3.3.1 Mathematical analysis of the Zero moment point

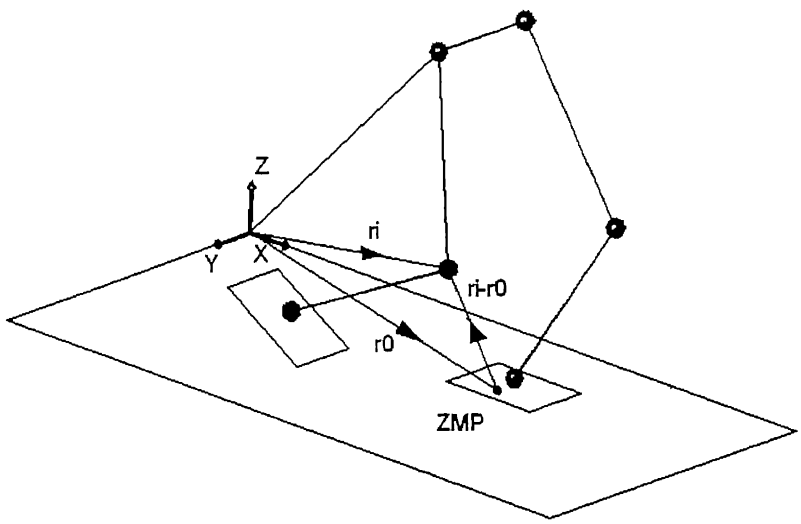


Figure 3.6. Mathematical analysis of dynamically stable biped robot.

The calculation of ZMP is explained below.

A reference frame is fixed at the corner of the place on which the robot walks, as shown in the Figure 3.6. If the position of the ZMP is 'r_0' and the position of i^{th} link's C.G. is 'r_i'. Then the position of the i^{th} link's C.G. with respect to the ZMP will be "$r_i - r_0$".

The moment of the mass 'M_i' about the ZMP will be

$$[r_i - r_0] \times [F_i] + [T_i] = [0] \qquad \text{---------- (3.1)}$$

Where F_i and T_i are the force and torque acting on the i^{th} link. And symbol 'X' shows cross product between two vectors. The above equation can be written in the following way:

$$\begin{bmatrix} x_i - x_0 \\ y_i - y_0 \\ z_i - z_0 \end{bmatrix} \times \begin{bmatrix} F_{ix} \\ F_{iy} \\ F_{iz} \end{bmatrix} + \begin{bmatrix} T_{ix} \\ T_{iy} \\ T_{iz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{---------- (3.2)}$$

Taking the cross product the equation will be

$$\begin{bmatrix} (y_i - y_0)F_{iz} - (z_i - z_0)F_{iy} \\ (z_i - z_0)F_{ix} - (x_i - x_0)F_{iz} \\ (x_i - x_0)F_{iy} - (y_i - y_0)F_{ix} \end{bmatrix} + \begin{bmatrix} T_{ix} \\ T_{iy} \\ T_{iz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{---------- (3.3)}$$

This equation gives the relation between the coordinates of the ZMP for only one mass. The equation for all 'n' links of the biped is calculated by summing all the masses.

$$\begin{aligned} \sum_{i=1}^{n} [(y_i - y_0)F_{iz} - (z_i - z_0)F_{iy}] + \sum_{i=1}^{n} T_{ix} &= 0 \\ \sum_{i=1}^{n} [(z_i - z_0)F_{ix} - (x_i - x_0)F_{iz}] + \sum_{i=1}^{n} T_{iy} &= 0 \\ \sum_{i=1}^{n} [(x_i - x_0)F_{iy} - (y_i - y_0)F_{ix}] + \sum_{i=1}^{n} T_{iz} &= 0 \end{aligned} \qquad \text{---------- (3.4)}$$

Since we want to get the position of ZMP in X-Y plane, this means that putting $Z_0 = 0$ in the above equation we get the X_0 and Y_0 coordinates of ZMP as follows:

$$x_0 = \frac{-\left[\sum_{i=1}^{n}\left\{z_i F_{ix} - x_i F_{iz} + T_{iy}\right\}\right]}{\left[\sum_{i=1}^{n} F_{iz}\right]} \qquad \text{---------- (3.5)}$$

$$y_0 = \frac{\left[\sum_{i=1}^{n}\left\{y_i F_{iz} - z_i F_{iy} + T_{ix}\right\}\right]}{\left[\sum_{i=1}^{n} F_{iz}\right]}$$

Equation (3.5) gives us the position of the ZMP on X-Y plane. This point must lie inside the supporting foot perimeter for stable walk.

F_{ix}, F_{iy}, and F_{iz} can be calculated from Newton's second law as:

$$M_i \times \ddot{x}_i = F_{ix}$$
$$M_i \times \ddot{y}_i = F_{iy} \qquad \text{---------- (3.6)}$$
$$M_i \times \ddot{z}_i = F_{iz} - M_i \times g$$

Any dynamic biped usually has 8-12 degrees of freedom. Therefore by using equation (3.5) and inverse kinematics, the complete configuration of the biped can not be derived. Therefore researchers have proposed an approach based on the selection of those trajectories of hip and leg which keep the ZMP within the supporting foot region.

The leg trajectory is the path which the ankle of the leg follows while in air, similarly hip trajectory is the path which the hip joint follows while the leg is in air. The hip and leg trajectories are shown in Figure 3.7.

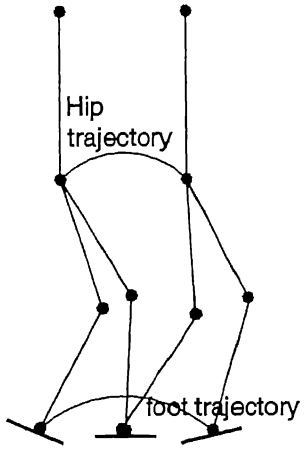


Figure 3.7. Hip and leg trajectories.

The foot trajectory depends on the step length and the maximum height of the ankle joint in air. The step length is 'sl', the height is 'mh' and three points are known on the trajectory, i.e. starting point, middle point and end point.

Let the origin be located at the ankle of the supporting foot, then the starting point 'P1', middle point 'P2' and the end point 'P3" are as given by (Figure 3.8)

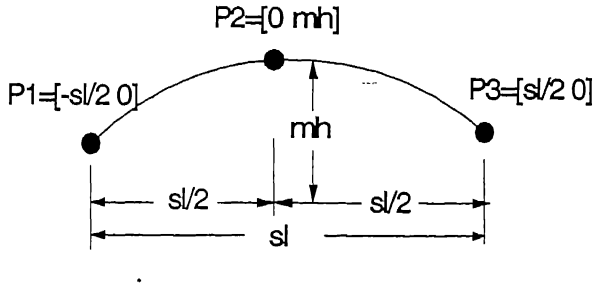


Figure 3.8. The ankle trajectory.

P1= [-sl/2 0]

P2= [0 mh]

P3= [sl/2 0]

Connecting these three points by a third order polynomial curve, the foot trajectory was calculated. The hip trajectory must be such that the ZMP will lie inside the foot region. Therefore the hip trajectory was taken as a variable. If the maximum height of the hip is 'MH_H' and the vertical movement of the hip is 'ΔMH_H', and as shown in Figure 3.9 P1, P2 and P3 are given by

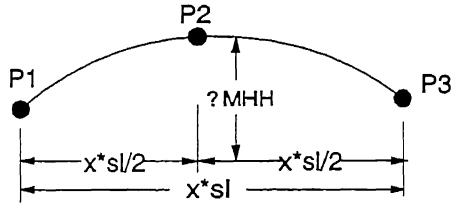


Figure 3.9. The hip trajectory.

P1= [-x*sl/2 MH_H-ΔMH_H]

P2= [0 MH_H]

P3= [x*sl/2 MH_H]

Connecting these points by a third order polynomial gives the hip trajectory. An iterative method was used, in which first 'x' was taken and then reduced until the ZMP was inside the foot region. As the position of the ankle of the supporting leg, position of the ankle of swinging leg and hip position are known; the configuration of all the biped links was computed using inverse kinematics. For the computed configuration the position of ZMP was again recalculated and if it does not lie inside the safe region of the supporting foot, then the hip trajectory is changed slightly. Figure 3.10 shows the stable area of the supporting foot. The Gait is stable if the ZMP lies within ±10mm of the centre line of the foot.

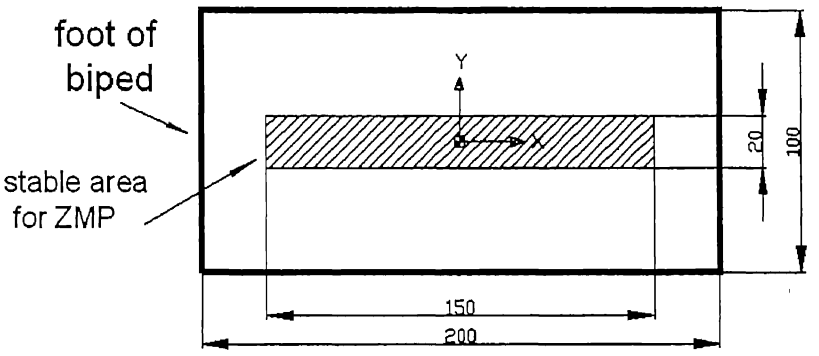


Figure 3.10. Stable area of foot of biped robot.

If the hip and leg trajectories are known with respect to time, then the positions of the ankle of the supporting leg, ankle of the swinging leg and hip joint can be computed.

Ankle of supporting leg D1=[x1 0 z1]
Ankle of swinging leg D2=[x6 y_w z6]
Hip joint of supporting leg D3=[x3 0 z3]

Figure 3.11 shows the axis and the values of angles for a given configuration. It has been assumed that the robot has its supporting foot on the x axis. Y_w is the distance between the two hip joints in y direction in normal standing position. At first the balance in y direction is not considered. Therefore x and z coordinates of the hip joint of the swinging leg will also be the same as that of the supporting leg. Hence D3=[x3 y_w z3]

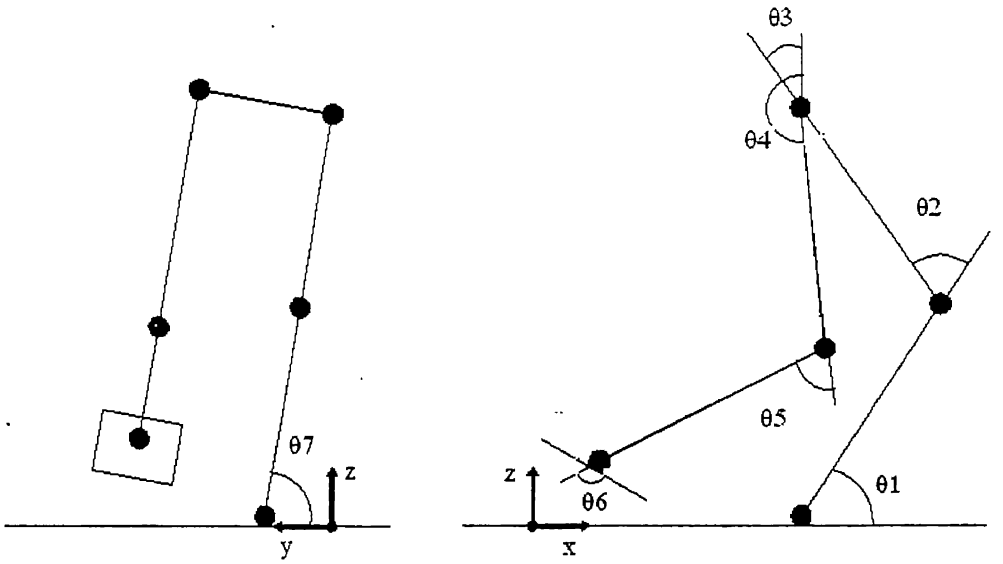


Figure 3.11. Angles of the links.

Let the length of the link above the knee joint be L1, the length of the link below the knee joint be L2 and the angle of the lower link with horizontal be Θ_1, then

$$\theta_1 = \alpha_1 - \beta_1 \;; \quad \text{if } x3 \geq x1$$
$$\theta_1 = \Pi + \alpha_1 - \beta_1 \;; \quad \text{if } x3 < x1 \qquad \text{---------- (3.7)}$$

Where,

$$\alpha_1 = \tan^{-1}\left[\frac{(z3 - z1)}{(x3 - x1)}\right]$$
$$\beta_1 = \cos^{-1}\left[\frac{L2^2 + L3^2 - L1^2}{2 \times L2 \times L3}\right] \qquad \text{---------- (3.8)}$$
$$L3 = \sqrt{\left[(x1 - x3)^2 + (z1 - z3)^2\right]}$$

Hence the position of the knee joint of the supporting leg is given by:

$$x2 = x1 + L2 \times \cos(\theta_1)$$
$$y2 = 0 \qquad \text{---------- (3.9)}$$
$$z2 = z1 + L2 \times \sin(\theta_1)$$

The angle Θ_6 can be computed as:

$$\theta_6 = \alpha_6 - \beta_6 \; ; \qquad \text{if } x4 \geq x6$$
$$\theta_6 = \Pi + \alpha_6 - \beta_6 \; ; \qquad \text{if } x4 < x6 \qquad \text{---------- (3.10)}$$

Where,

$$\alpha_6 = \tan^{-1}\left[\frac{(z4 - z6)}{(x4 - x6)}\right]$$
$$\beta_6 = \cos^{-1}\left[\frac{L2^2 + L3^2 - L1^2}{2 \times L2 \times L3}\right] \qquad \text{---------- (3.11)}$$
$$L3 = \sqrt{\left[(x6 - x4)^2 + (z6 - z4)^2\right]}$$

Hence the position of the knee joint of the supporting leg is,

$$x5 = x6 + L2 \times \cos(\theta_6)$$
$$y5 = y_w \qquad \text{---------- (3.12)}$$
$$z5 = z6 + L2 \times \sin(\theta_6)$$

The ankle joint and the knee joint of the supporting leg are rotated to obtain the required hip joint position. The hip joint motor rotates only to keep the trunk vertical. The hip joint and the knee joint motors of the swinging leg are rotated to keep the ankle point on the trajectory. The ankle joint motor will only rotate to keep the foot of the swinging leg always horizontal. The angle of rotation of each motor with respect to the horizontal is given by:

$$\theta_2 = \Pi + \tan^{-1}\left[\frac{(z3 - z2)}{(x3 - x2)}\right]$$
$$\theta_3 = \frac{3 \times \Pi}{2} - \tan^{-1}\left[\frac{(x3 - x2)}{(z3 - z2)}\right] \qquad \text{---------- (3.13)}$$
$$\theta_4 = \frac{3 \times \Pi}{2} - \tan^{-1}\left[\frac{(x4 - x5)}{(z4 - z5)}\right]$$
$$\theta_5 = \tan^{-1}\left[\frac{(z5 - z6)}{(x5 - x6)}\right]$$

Up to this point the balance in frontal plane has been neglected and therefore the Y coordinate of the ZMP is $y_w/2$. For balancing the ZMP must lie inside the supporting foot region, therefore two motors are provided at each ankle, to bend the robot in the frontal plane (Figure 3.12). By rotating the motor of the supporting leg ankle, the ZMP is brought inside the supporting foot region. However in order to keep x and z coordinates of the ankle of the swinging leg unchanged, the values of Θ_4, Θ_5 and Θ_6 are changed, such that the ankle of the swinging leg is on the desired leg trajectory.

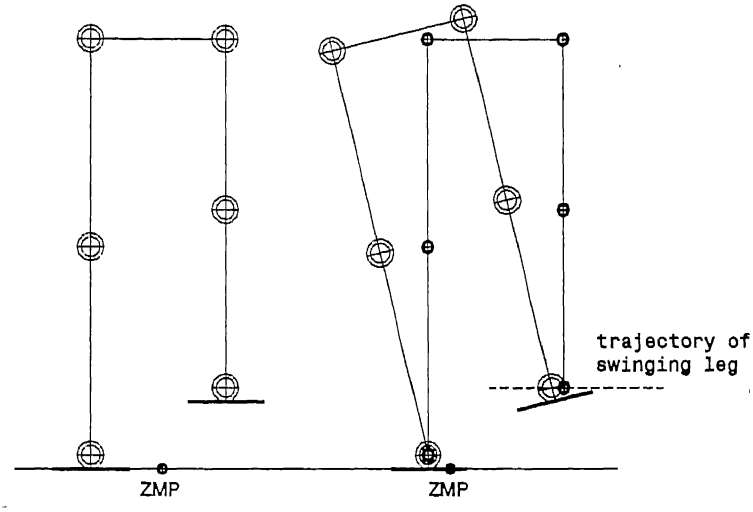


Figure 3.12. Bending in frontal plane to keep ZMP in foot area.

3.3.2 Simulation of Dynamic walking

One MATLAB simulation has been developed for generating the required configurations of the dynamic biped for walking. The inputs to the simulations are:

a) The step length
b) The walking speed of biped
c) Length of all the links
d) Weight of all the links and actuators
e) Maximum change in height of the leg and hip during a step

The outputs of the simulation are:

a) The angles of all the links.
b) Position of the ZMP for each configuration

c) Graphical animated view of dynamic walking

d) Figure showing foot placements and the ZMP inside the foot region

Other than this simulation, one MATLAB graphical user interface has also been developed in which the view of the biped can be selected from a pop-up-menu and the number of steps can be given from an edit box.

3.3.3 Results of Dynamic Walking

The results shown in the Figures 3.13, 3.14, 3.15 and 3.16 were obtained by the following input data.

Step length = 20cm.

Walking speed=10cm.

Length of each link = 22 cm.

Weight of each link = 1kg.

Weight of each actuator = 1kg.

Change in height of hip and ankle = 1.5cm and 3.5cm respectively.

The Figure 3.13 shows the side view of the simulation results of dynamic walking in single support mode.

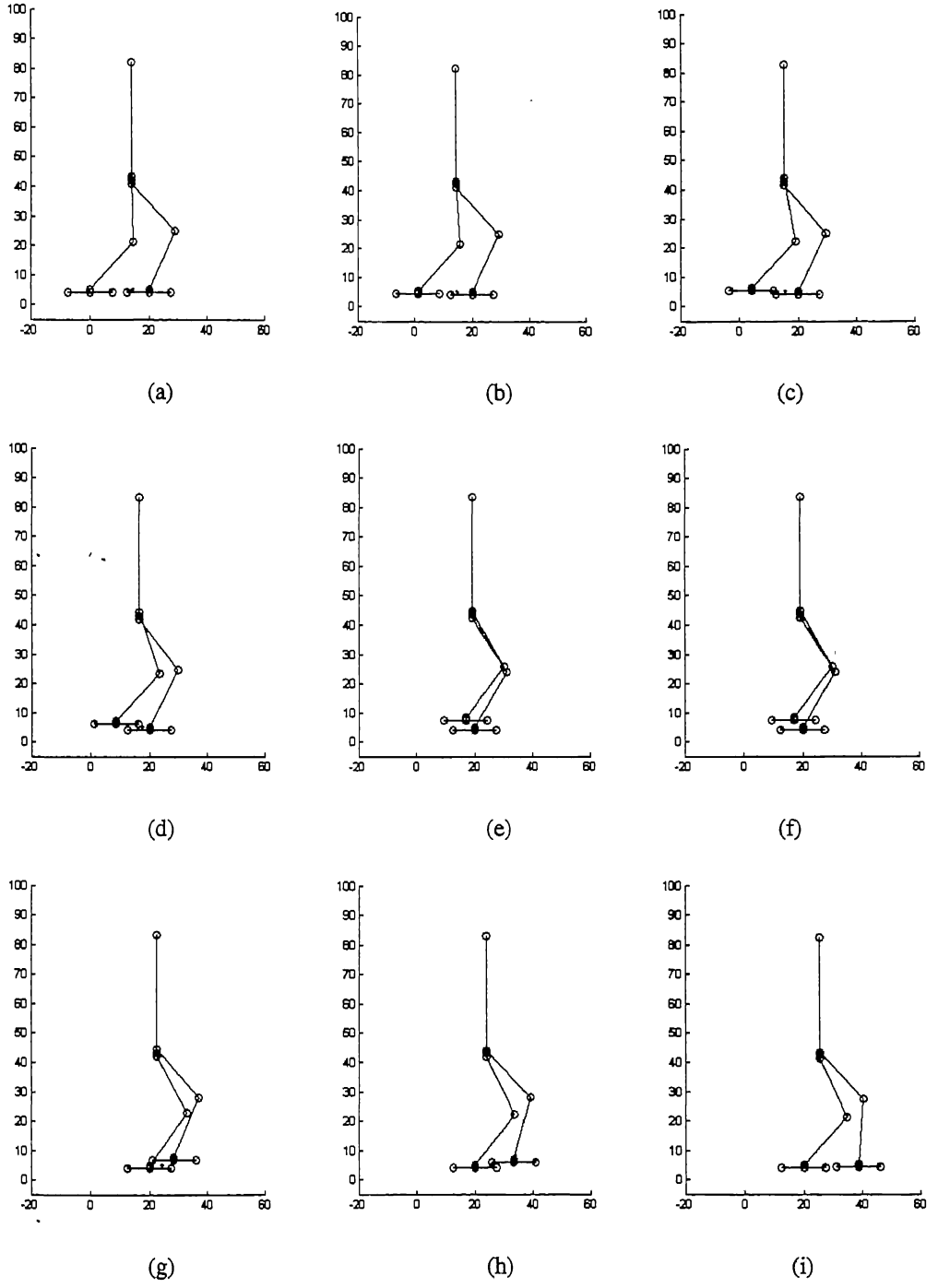


Figure 3.13. Side view of the dynamic walking in single support phase.

The Figure 3.14 shows the front view of the simulation results of dynamic walking in single support.

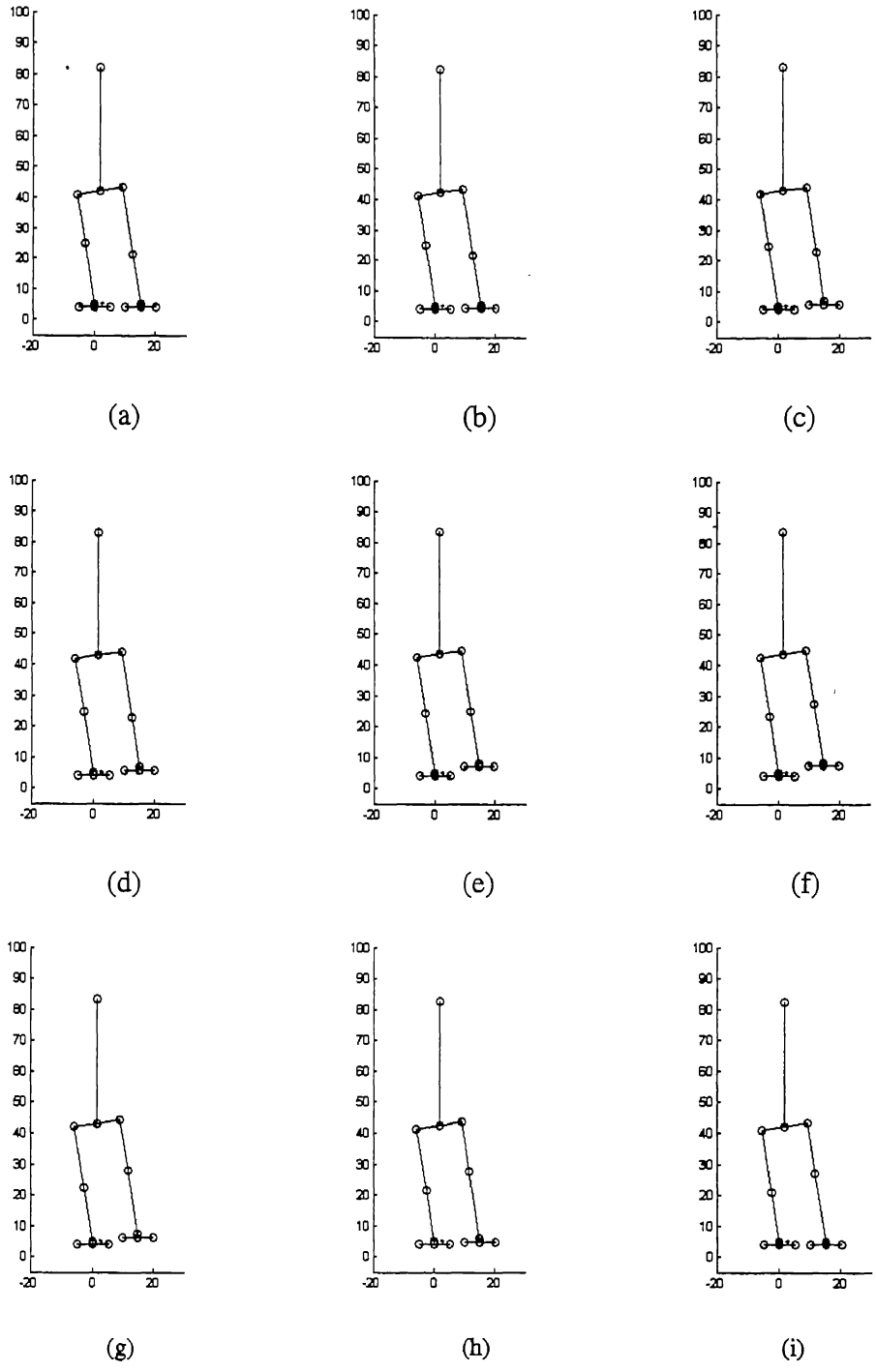


Figure 3.14. Front view of dynamic walking in single support phase.

The Figure 3.15 shows the side view and front view of the simulation results of dynamic walking in double support.

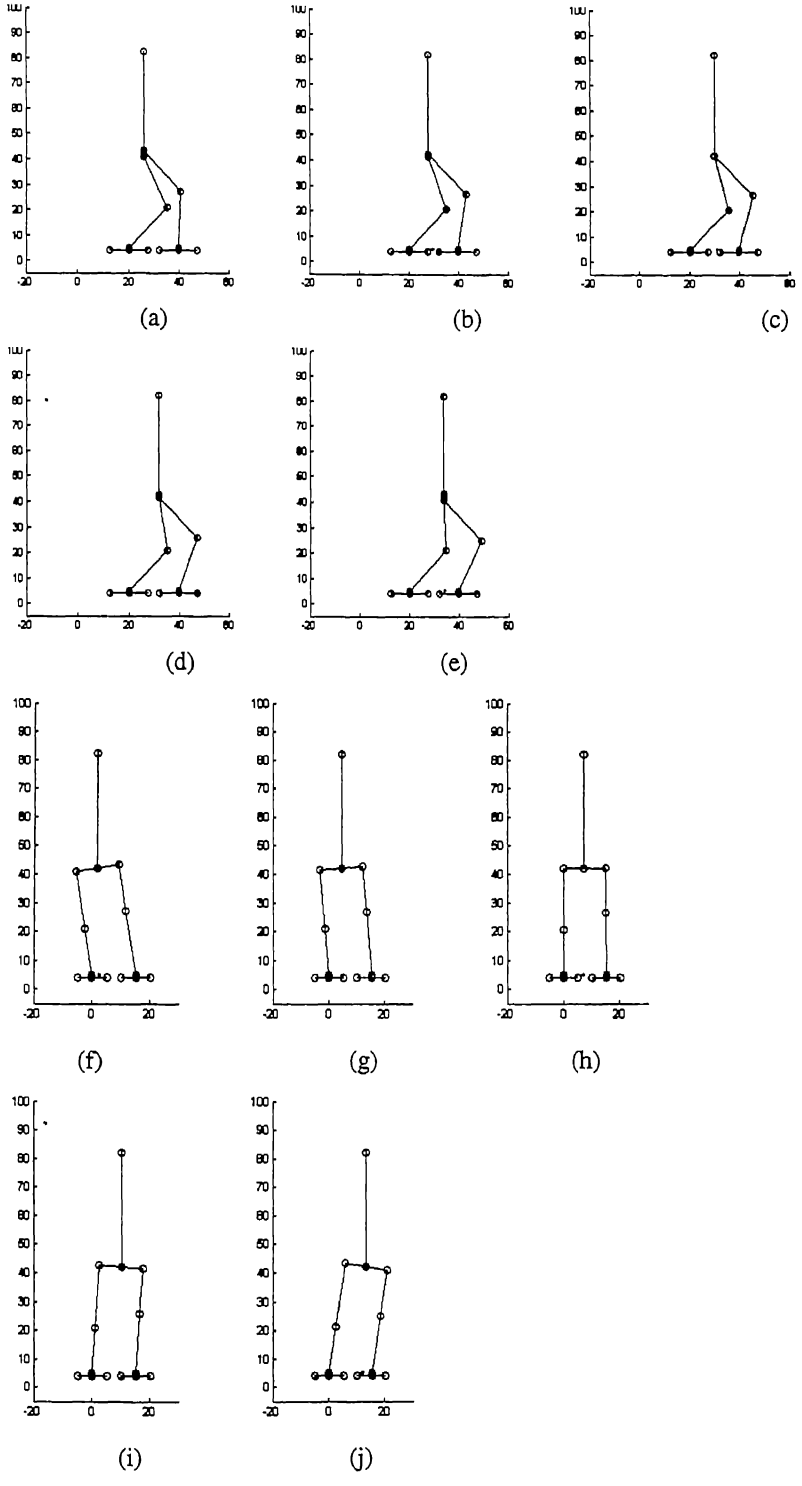
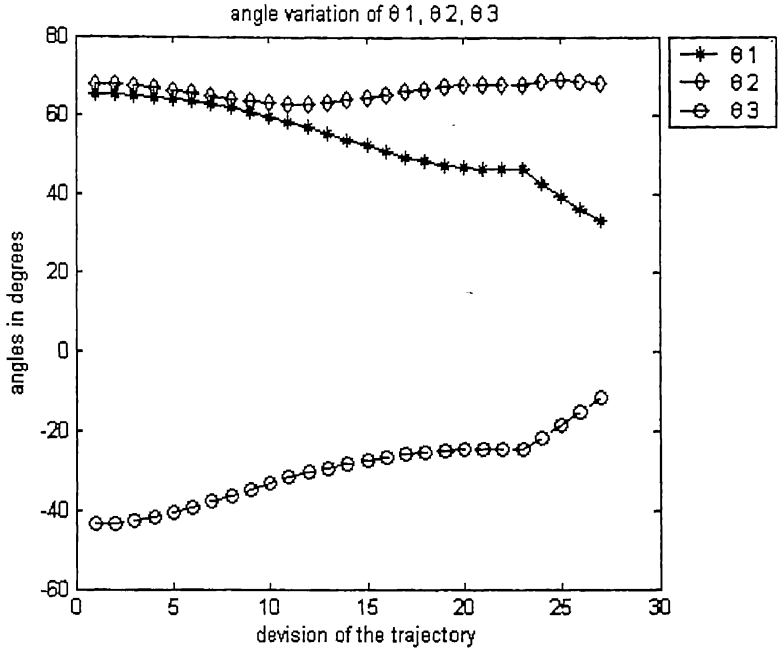
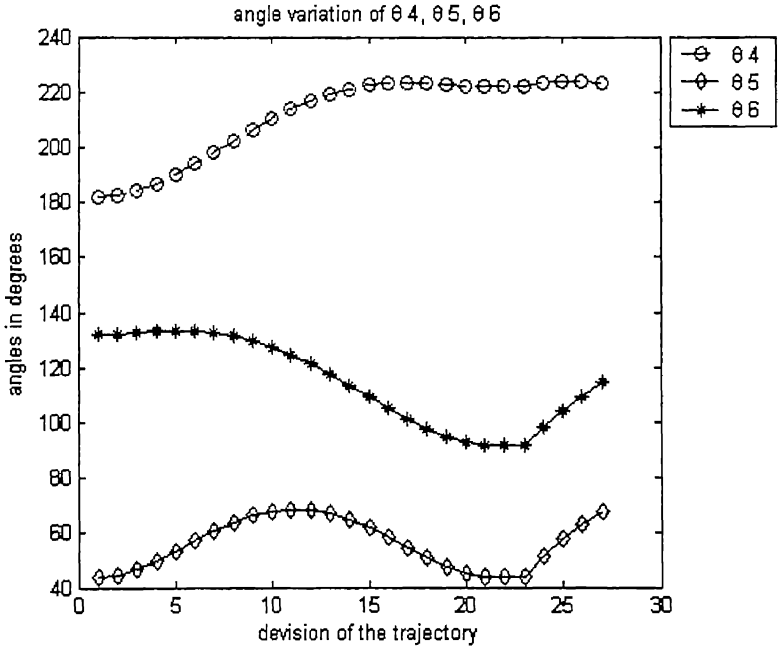


Figure 3.15. Front and side view of the dynamic walking in double support.

Figure 3.16 shows the variation of all joint angles for single step. (the angles are as designated in Figure 3.11)



(a)



(b)

Figure 3.16. Angle variation of the joints.

Figure 3.17 shows the results of the MATLAB graphical user interface. Figure 3.16 (a) shows the graphical user interface itself. Figure 3.16 (b), (c) and (d) are the results of isometric, front and side view respectively.

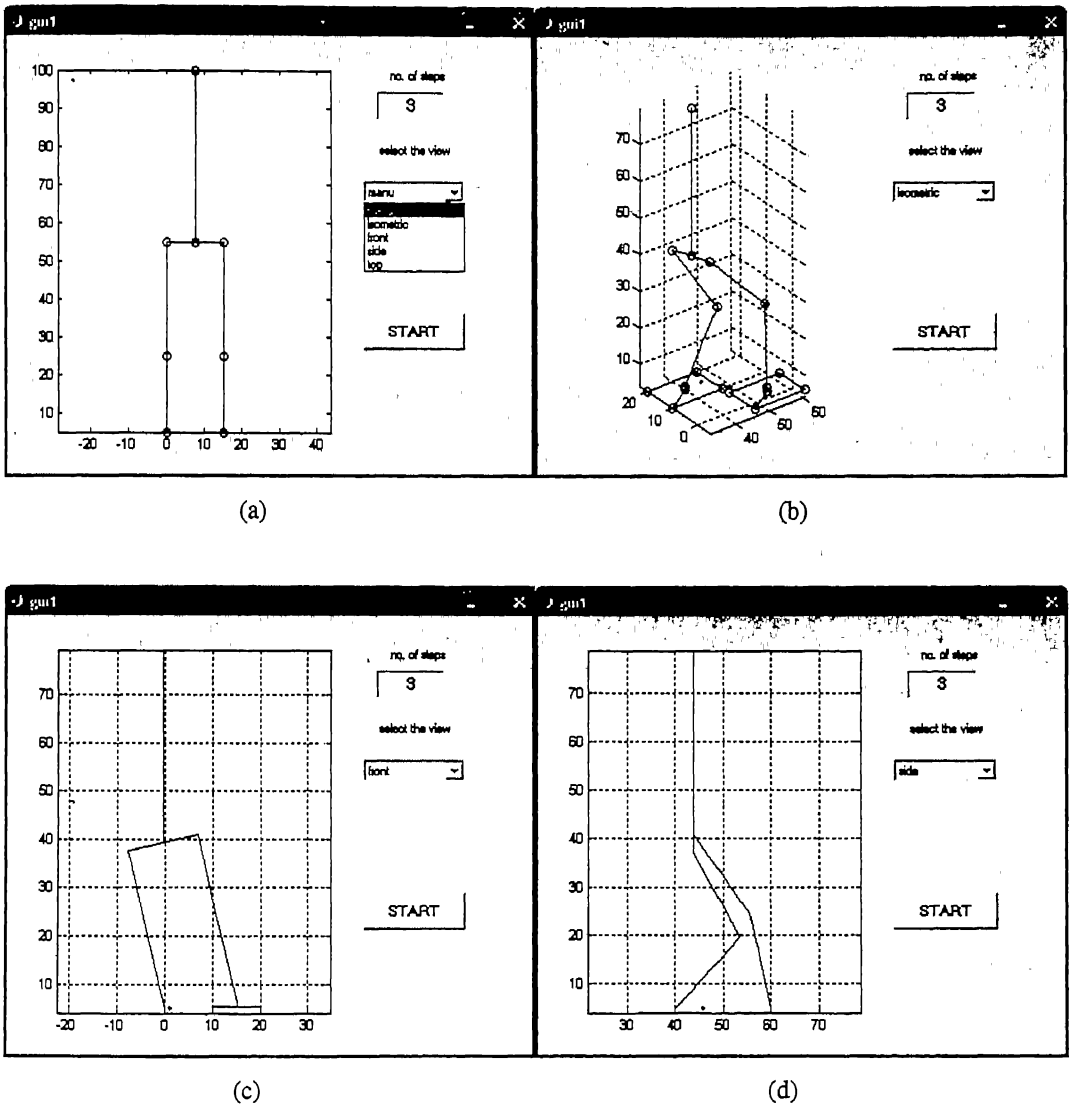


Figure 3.17. The results of the graphical user interface.

Chapter 4

Design of statically stable biped robot

The main objective of this chapter is to design a suitable structure for the statically stable biped robot such that the biped can support all the forces and moments acting on the members. Four actuators provide all the required torques and the motor controller controls the joints. In order to follow a straight line, a statically stable robot requires only one actuator (if the legs are coupled). In case the legs are independent, then two actuators are required. Two more actuators are required, one at each ankle joint, for turning the torso of the robot. Hence for walking in a straight line and turning, a total of four actuators are required. The number of actuators required are much less than that for a dynamically stable biped robot, which is at least ten. The mechanical design of a humanoid robot requires a complex tradeoff between form, function, power, weight, cost and manufacturability. The main points considered while designing the robot were:

a) The minimum number of motors required for satisfying the requirements of trajectory following.
b) It should be statically stable at all times during gait.
c) Cost should be a minimum.
d) The structure should be same as that of a small child of height about 1.2m.

Figure 4.1 shows the statically stable biped robot.

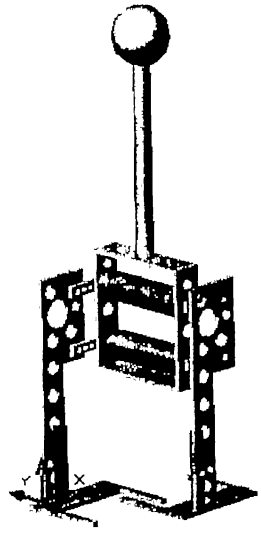
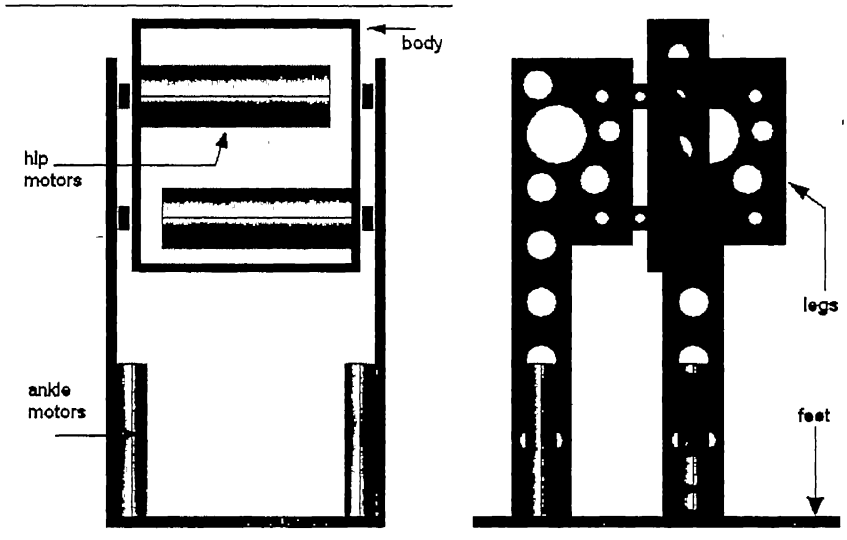


Figure 4.1. Statically stable biped robot.

The basic details of the biped robot are as explained in [37]. In order to use the minimum number of actuators the designed robot does not have any knee joint. It is actuated at the hip joint by a four bar mechanism and the legs move in circular fashion in order to advance to the next step. The turning motion of the robot is provided by the motors at the ankle. When the robot is on a single foot stance the ankle motor of the foot on the ground can rotate, which rotates the torso. Using this procedure for trajectory following the robot can walk straight or turn. The two legs are independently connected to a four bar mechanism which is actuated by a geared DC servomotor. The number of motors can be further reduced if one motor actuates both the hip joints. However, independent drives were provided because two legs require independent motion for trajectory following, climbing steps, dancing etc.

4.1 Basic mechanical design



a) Lateral view b) sagittal view

Figure 4.2. Side view and front view of the statically stable biped robot.

The basic design of the statically stable biped robot is as shown in Figure 4.2 in which the side view (sagittal plane) and front view (lateral plane) is shown. It has two legs, one body, two large feet, two motors are attached to the body for forward motion and two motors are attached to the ankle joint for turning of the torso. The dimensions of all the members have been minimized to reduce the total weight of the robot. The basic

size of the humanoid is the same as that of a small child. The legs have been designed to traverse a distance of 200 mm per step, in 2.5 seconds. The foot dimensions have been selected to ensure that the robot is stable at all times during gait. The shape and the dimension of the feet have been decided to avoid foot interference during gait. As shown in Figure 4.2, during gait there are two planes in which the robot moves:

a) The motion in the lateral plane, in which case the projection of the CG of the robot remains fixed at the center of the torso, and is within the frame of the foot support polygon.

b) In the sagittal plane, as the robot moves to the front or back the CG moves along the trajectory followed by the center of the torso. As this also falls with in the perimeter of the foot, the robot is always balanced even when it is on one-foot stance.

c) The body is in an upright position at all times during gait.

4.1.1 Design of the leg

The main consideration for the design of the leg is that it must be able to withstand the load of the body, driving links, motors and torso. The driving four bar links are attached to the leg by means of thrust bearings. Several holes have been made to reduce the weight of the leg. The weight of one leg is about 5kg and the basic dimensions of the leg are as shown below in Figure 4.3. The section of the leg is 13mm thick and it is made of heat treated Aluminum alloy.

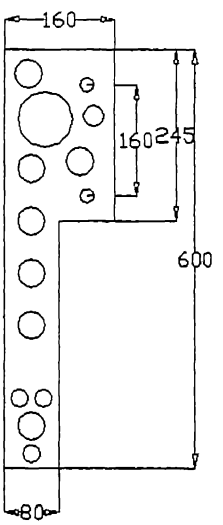


Figure 4.3. The basic dimensions of each leg.

4.1.2 Design of the foot

The foot attached to the leg, has to be such that the center of gravity will always lie inside its perimeter. The projection of the center of gravity has two coordinates as the Z coordinate is always zero. The x-coordinate (X_{cg}) is in the direction of walking and the other is perpendicular to the walking direction (Y_{cg}). In our model for straight walking X_{cg} moves by 200mm and Y_{cg} remains constant at the middle position between the two legs. Therefore, the size of the feet in x direction must be more than 200mm and in y direction it must be more than half the distance between the two legs. As the robot moves from one foot onto another, the CG also moves from one foot to the other. Therefore both the feet must have a common area where the CG will be in the changing over phase, as shown in Figure 4.4 below. The CG moves from -50mm to +150mm relative to the ankle motor axis of the stance leg as shown in Figure 4.4 by a dotted line. Therefore the position of the foot is not symmetric in x direction. The thickness of the foot is 13mm and its weight is about 1kg. The material of the foot is heat treated aluminum alloy.

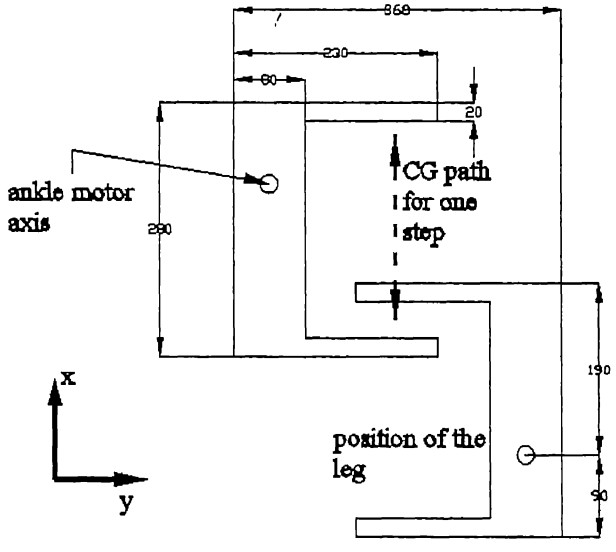


Figure 4.4. The basic dimensions of the feet of the statically stable biped robot.

4.1.3 Design of body

The body of the robot is connected to both of the legs by a four bar mechanism. The legs are rotated by the motors via the four bar mechanisms. The body also supports the weight of the motors and the upper trunk. Therefore the main consideration for the

design of the body is its strength to carry all the load of the motors, torso, leg while it is on one foot stance. The total weight of the body along with the motors and torso is about 9 Kg. The basic dimensions of the body are as shown in Figure 4.5.

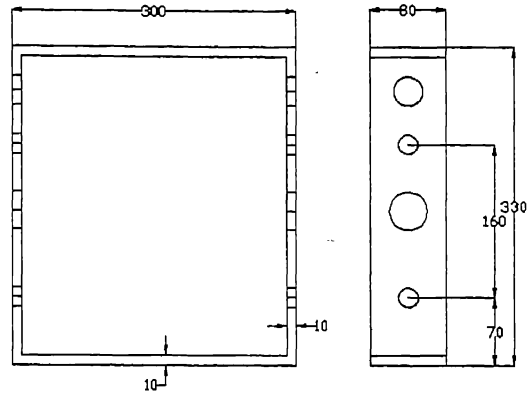


Figure 4.5. The dimensions of the body of the statically stable biped robot.

4.1.4 Design of four bar links

The four bar links connecting the motors to the legs are such that the distance between the motor shaft and leg is 100mm. Since motion is being transferred through the link which is made of aluminum alloy and the shaft of the motor is made of high carbon steel. Steel inserts have been used in connecting the motor with four bar links. In this case also several holes have been provided to reduce its weight. The basic dimensions of the body are as shown in Figure 4.6.

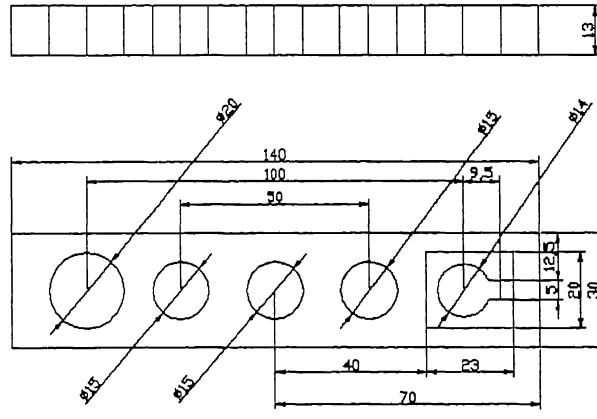


Figure 4.6. The dimensions of the links used in statically stable biped robot.

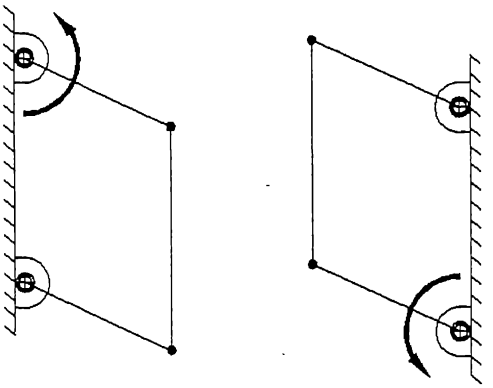


Figure 4.7. A four bar mechanisms.

Two parallel four bar mechanisms have been used for advancing the leg for moving the robot forward or backward. These mechanisms connect the two legs to the body and one of the links is actuated by the hip motor. Four bar mechanism has been used as they can move the legs upward and also move the body forward with the same motion. The basic working principle of the four bar mechanism is as shown in Figure 4.7.

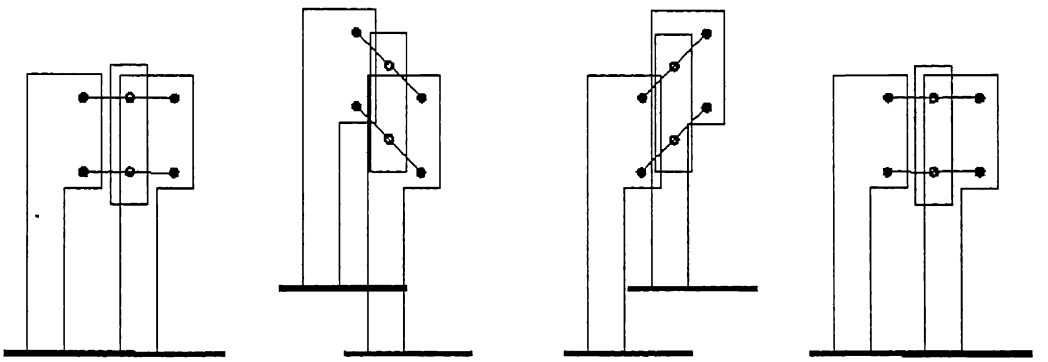


Figure 4.8. Working of four bar mechanisms in statically stable biped.

Both the four bar mechanisms are actuated by two independent hip motors. One motor is attached to the upper link (connecting the body to the leg) and the other motor is attached to the lower link of other mechanism. The basic motion of the robot is as shown in Figure 4.8.

The forces acting on the legs, four bar links and body are as shown in Figure 4.9.

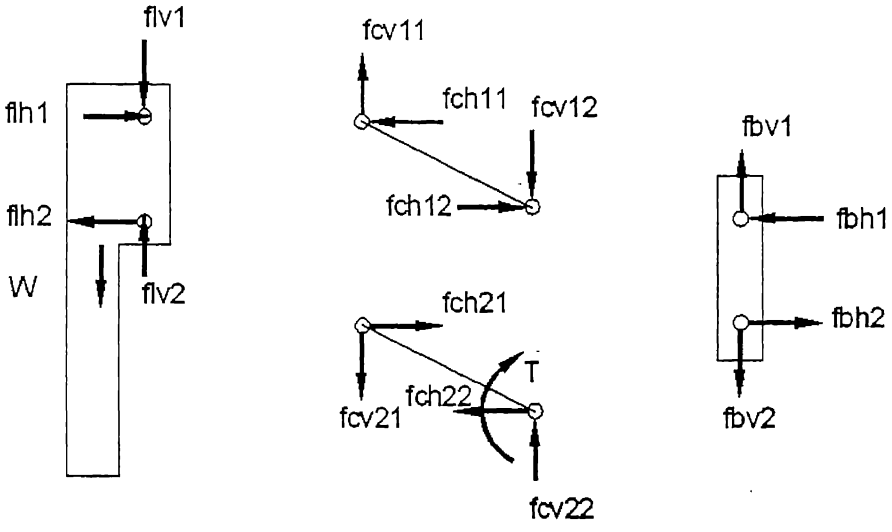


Figure 4.9. Forces on four bar mechanism.

Since the centre of gravity of each leg has an offset from the supporting points, a torque will be acting on the leg due to its own weight which will try to rotate the leg when it is in air. A reactive torque is generated by two horizontal reaction forces (flh1 & flh2) coming from the connecting links. Equal and opposite forces will act on the links (fch11 & fch21), which will generate torques on both the links. The link connected to the motor will get actuating torque from motor (T). The other link experiences one horizontal and one vertical reaction (fch11 & fcv11) from the leg. The torque generated by fch11 will be cancelled by the torque generated by fcv11. The resultant of all the forces and torques ensures that the leg will only translate, not rotate, and move forward parallel to the body.

The Four bar mechanism has an inherent problem of having a singular position when the four links are in line with each other. When the links connecting the body to the legs are vertical, the vertical force on the link (fcv11) can not generate any torque as it generates only a tension in the link, therefore the leg will experience a reverse torque because of the horizontal force acting on the link (fch11). At this position the four bars tends to become 'stuck'. In order to bring it out form this position a very high force is required which causes an undesired jerk in the mechanism. In order to prevent the

inversion of the leg and to rotate it continuously at the singular position, a positive drive using chain and sprocket have been provided. A chain drive which connects the motor shaft to the opposite link which is not directly connected to the motor shaft has been used. This ensures that the torque of the motor is given to both the links and therefore the link at the singular position will not rotate in the opposite direction or cause a severe jerk.

4.2 Force and torque calculation

All the dynamic forces have been neglected for the calculation of the forces and torques; because the robot moves very slowly (it takes one step in about 3 seconds). The following data has been used for the calculation:-

Weight of each leg = 1.8 kg

Weight of each foot = 1 kg

Body weight (without motor mounting) = 2.6 kg

Weight of each four bar link = 0.1 kg

Weight of head and torso = 9 kg

Weight of each hip motor = 2.5 kg each

Weight of each ankle motor = 1.5 kg each

When one foot is on the ground and the other is in air, the weights of the links in air will be propagated to the foot on the ground. The leg in air has to support the tensile forces and bending moments of the foot, whereas the leg on the ground will have the compressive forces and the bending moments of the whole robot. The shafts of the hip motors will experience bending stresses as well as shearing stresses. The shaft of the hip and ankle motor will have tensile as well as bending stresses. The four bar links (connectors) will have bending as well as shearing stresses. The calculation of all the forces and moments are given below.

The weight of the foot in air will create two moments in two perpendicular directions (Ms1 & Ms2) and one tensile force on the motor shaft (Ts) of the leg in air. For calculating the forces and moments, the position of the centre of gravity is first calculated.

$$X_{CG} = \frac{[280*230*150-150*240*150-pi*20\hat{}2*130-pi*20\hat{}2*60+pi*20\hat{}2*50-pi*5\hat{}2*0]}{[280*230-150*240-pi*20\hat{}2*3-pi*5\hat{}2]}$$

$X_{CG} \cong 51mm$

$$Y_{CG} = \frac{[280*230*75-150*240*115-pi*20\hat{}2*0-pi*20\hat{}2*0-pi*20\hat{}2*0-pi*5\hat{}2*90]}{[280*230-150*240-pi*20\hat{}2*3-pi*5\hat{}2]}$$

$Y_{CG} \cong 28mm$

Where X_{CG} and Y_{CG} are the coordinates of CG in x and y direction respectively, Figure 4.10 shows the position of the centre of gravity about the ankle motor axis.

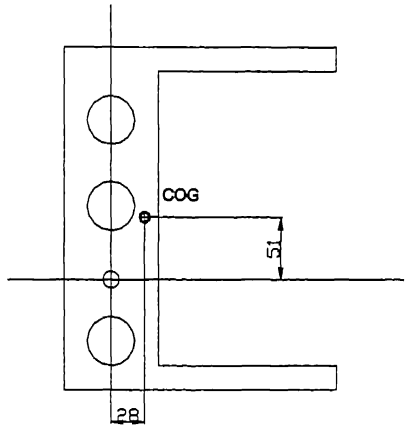


Figure 4.10. Position of CG on foot.

The weight of the foot is 1 kg, therefore the moments and forces were calculated as:

$$Ts = 1 \times 9.81\,N \quad \text{----------(4.1)}$$

$$Ms1 = -Mf1 = -1 \times 9.81 \times 51\,Nm$$

$$\cong -500\,Nmm\,(clockwise) \quad \text{----------(4.2)}$$

$$Ms2 = -Mf2 = -1 \times 9.81 \times 28\,Nm$$

$$\cong -275\,Nmm\,(clockwise) \quad \text{----------(4.3)}$$

Figure 4.11 shows the directions of moments and forces acting on the foot and the leg.

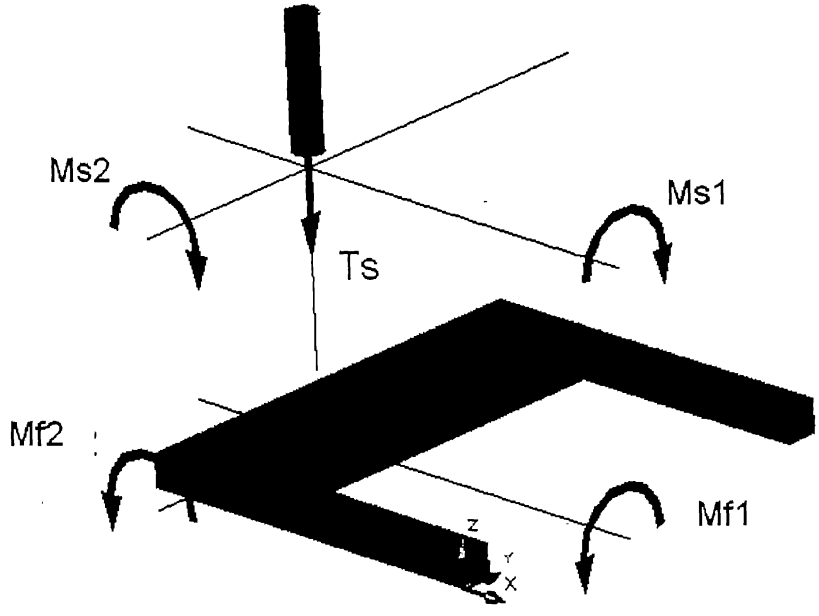


Figure 4.11. Forces and moments acting on the foot.

The weight of the leg and ankle motor will create some tensile stresses in the upper part of the leg. Since motor shaft has an offset (33.5mm) from the leg axis, the weight of the foot and motor will also create a moment. These forces will then be transferred to the four bar links. The two components of centre of gravity of the leg are as shown in Figure 4.12.

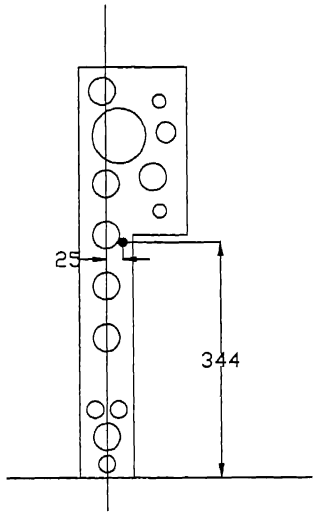


Figure 4.12. Position of CG in leg.

The moments and forces acting on the leg are as shown in the Figure 4.13.

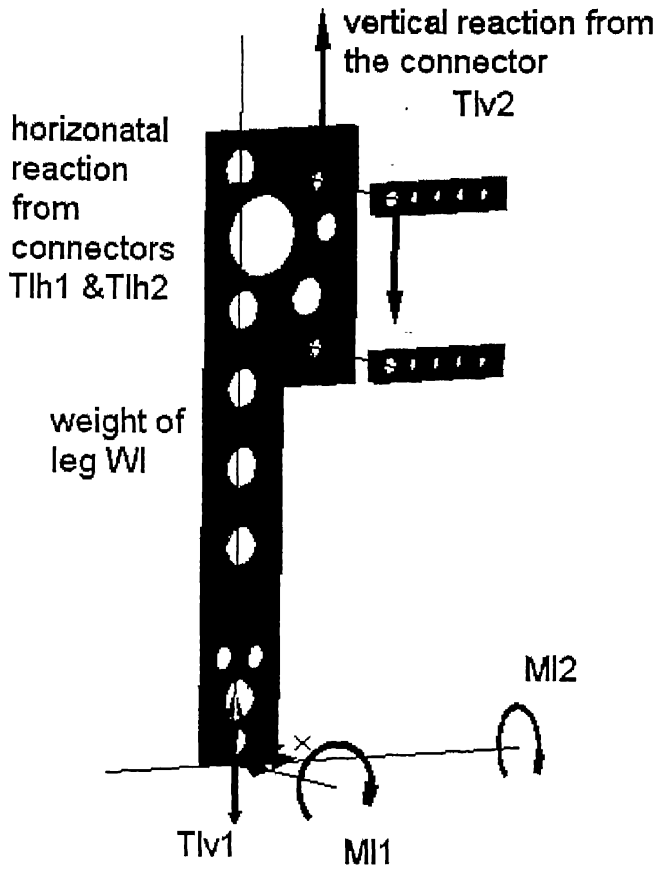


Figure 4.13. Forces and moments acting on the leg.

The force Tlv1 acting on the leg is due to the weight of the foot and the ankle motor. Since the leg is attached to two joints, there will be some reaction forces acting on the leg at the joints. The vertical actuation force can only be given by the connector which is fixed to the motor. Assuming that the upper link is fixed to the hip motor, the forces Tlv1, W (weight of the leg) and the reaction from the connectors (Tlv2) which are not collinear, will cause a moment on the leg, because of which it will tend to rotate. Since the connectors are attached to the leg by free rotating joints, the connectors will provide horizontal force reaction (Tlh1 & Tlh2) such that the moments due to the vertical forces will be balanced. The forces and moments are calculated as follows.

$$Tlv1 = Ts + Wa = 1 \times 9.81 + 1.5 \times 9.81 = 24.5N \quad \text{----------(4.4)}$$

$$Ml1 = Ms1 = 500Nmm(clockwise) \quad \text{----------(4.5)}$$

$$Ml2 = Ms2 + 2.5 \times 9.81 \times 33.5 \cong 1096Nmm(anticlockwise) \quad \text{----------(4.6)}$$

$$Tlv2 = Tlv1 + W = 24.5 + 1.8 \times 9.81 \cong 42N \quad \text{----------(4.7)}$$

$$Tlh2 = -Tlh3 = \left[\frac{Tlv1 \times 40 + W \times 16 - Ml1}{160}\right] = 4.76N \quad \text{----------(4.8)}$$

Forces Tlv2, Tlh1 and Tlh2 are the reaction forces from the connectors to the leg therefore the same forces in opposite direction will act on the hip motor shaft along with one bending moment (Mc2). The torque required for that hip motor is directly proportional to the forces on the link and the distance of the motor shaft from the link end. The forces acting on the link will be changing in a sinusoidal pattern, for one step and the distance moved by the leg. The Figure 4.14 shows the forces acting on each link of the four bar mechanism.

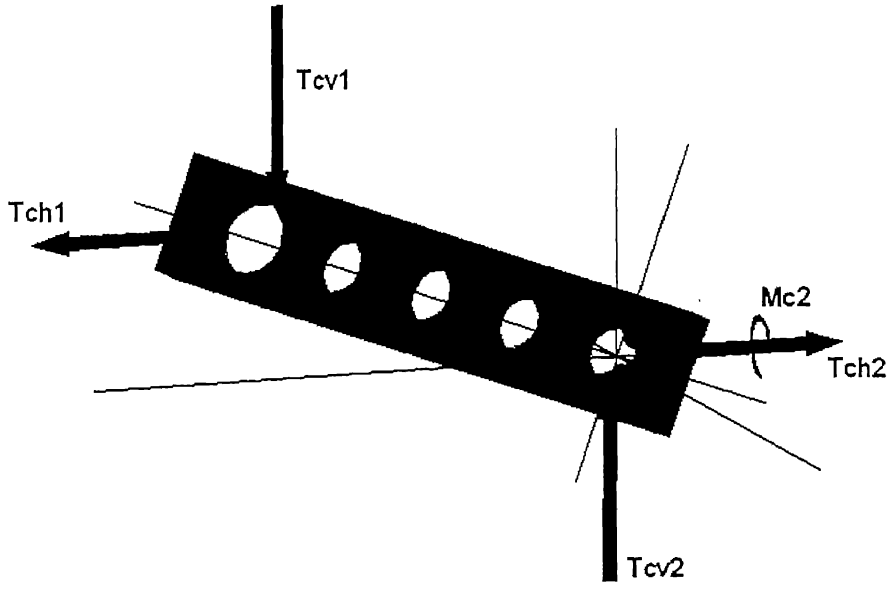


Figure 4.14. Forces and moments acting on the link.

$$Tcv1 = -Tlv2 = -42N \quad \text{----------(4.9)}$$

$$Tch1 = -Tlh1 = 4.76N \quad \text{----------(4.10)}$$

$$Mc2 = Ml2 + Tlv2 \times 17 = 1096 - 42 \times 17 = 362Nmm \quad \text{----------(4.11)}$$

The forces acing on the link will be transferred to the hip motor shaft, where it will exert a torque and shear forces. One bending moment will generate the bending stresses in the shaft. Figure 4.15 shows the forces and the torques acting on the shaft of the hip motor.

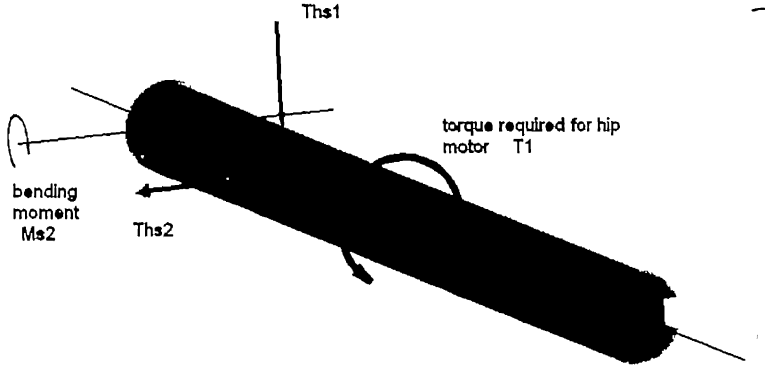


Figure 4.15. Shear forces, torques and moments on the shaft of hip motor.

The forces and the moments have been calculated as given below:

$$Ths1 = Tcv1 = 42N \quad \text{----------(4.12)}$$

$$Ths2 = Tch1 = 4.76N \quad \text{----------(4.13)}$$

$$Ms2 = Mc2 + Tcv1 \times 15 = 362 - 42 \times 15 = -268Nmm \quad \text{----------(4.14)}$$

All these forces will be same for each step but the torque requirement will be changing. Torque for any leg angle is given by:

$$T1 = Tcv1 \times 100 \times \cos(\theta) + Tch1 \times 100 \times \sin(\theta) \quad \text{----------(4.15)}$$

The Figure 4.16 shows the torque the torque variation on the hip motor attached to the leg in air, for one step.

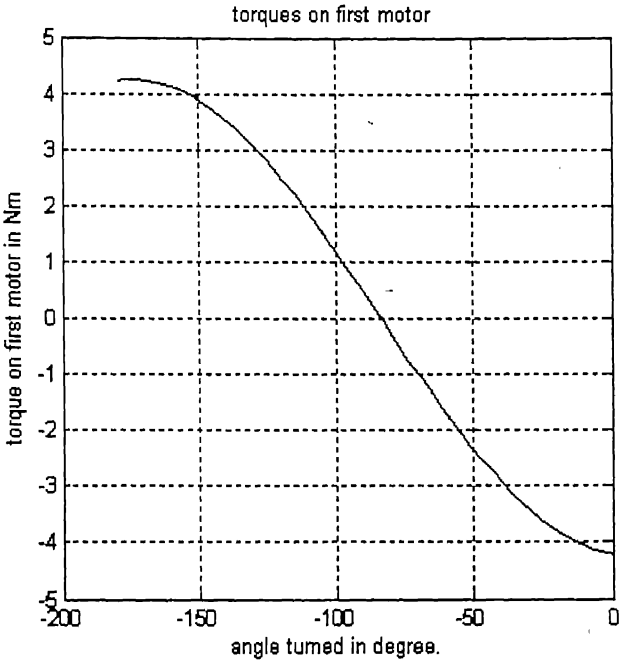


Figure 4.16. Torque acting on first hip motor.

Differentiating the torque equation (4.15) and equating it to zero, the angle for maximum torque was calculated. Similarly the angle for zero torque was calculated by equating the torque equation to zero.

$$T1 = 4200 \times \cos(\theta) + 476 \times \sin(\theta) = 0$$

$$\tan(\theta) = -4200/476$$

$$\theta = 1.6836\text{rad} = 96.4659^0 \qquad \text{----------(4.16)}$$

Torque will be zero when connector angle=96.4659°

$$dT1/d\theta = -4200 \times \sin(\theta_{max}) + 476 \times \cos(\theta_{max}) = 0 \qquad \text{----------(4.17)}$$

$$\tan(\theta_{max}) = 476/4200$$

$$\theta_{max} = 3.0287\text{rad} = -173.5313^0 \qquad \text{----------(4.18)}$$

$$T1(\theta_{max}) = 4.2269\text{Nm} \qquad \text{----------(4.19)}$$

The maximum torque required for the hip motor was found to be 4.2269Nm at Θ=173.5313°.

All the forces, torques and bending moments acting on the shaft of the hip motor will be transferred to the body of the biped. There will be two vertical forces acting on the body, one will be the force transferred by the shaft of the upper motor (Tbv1) and the second will be the weight of the body (Wb). These vertical forces are balanced by the vertical reaction force (Tbv2) coming from the shaft of the lower motor. Similarly all the horizontal forces (Tbh1 & Tbh2) are balanced by the horizontal reaction forces. The shaft of the lower motor generates a reactive bending moment which balances the bending moment transferred to the body by the upper motor shaft and the bending moment due to all vertical forces (Tbv1 & Wb). Since a torque is generated by the upper motor, this torque is transferred to the body (Mb1). This torque is balanced by two horizontal reaction forces (Tbh3 and Tbh4). Figure 4.17 shows the forces, torques and bending moments acting on the body.

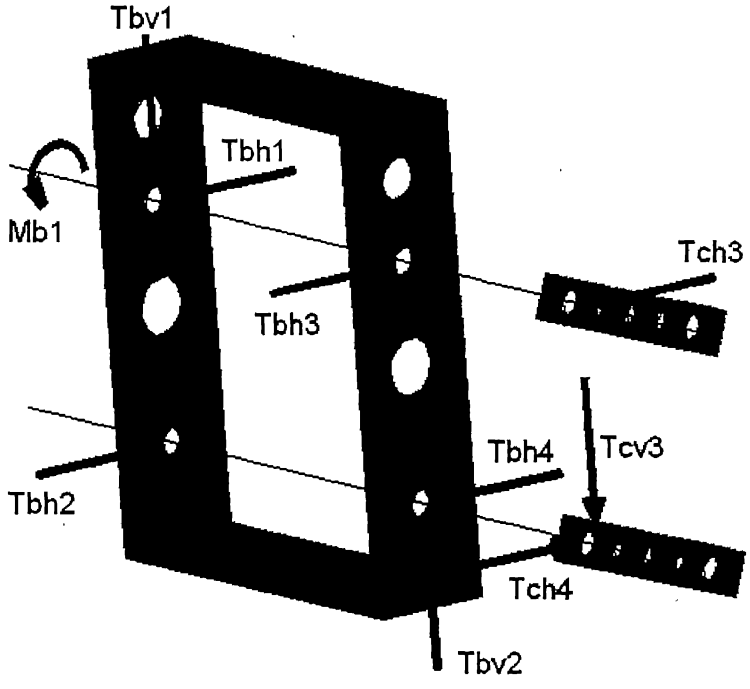


Figure 4.17. Forces and moments acting on the body.

The relation between the forces and moments are given by:

$Tbv1 = -Tcv2 = -42N$

$Tbv2 = -Tbv1 - Wb = 42 + 7.6 \times 9.81 = 116.5N$

$Tbh1 = -Tch2 = 4.76N$

$Tbh2 = -Tbh1$

$Mb1 = T1$ ---------- (4.20)

$$Tbh4 = -Tbh3 = \left[\frac{Mb1 + Tbh1 \times 160}{160}\right]$$

$Mb2 = Mc2 = -268Nmm$

$Mb4 = Mb2 + Tbv1 \times 290 + Wb \times 9.81 \times 145 = -23258Nmm$

All the forces acting on the body are balanced by the reaction forces coming from the shaft of the hip motor connected to the leg on the ground. Therefore the shaft will also carry the same load i.e. vertical force Tbv2 (downward) and horizontal force Tbh4. These forces are then transferred to the connectors which connect the leg to the body. The forces on the connectors are as shown in the Figure 4.18.

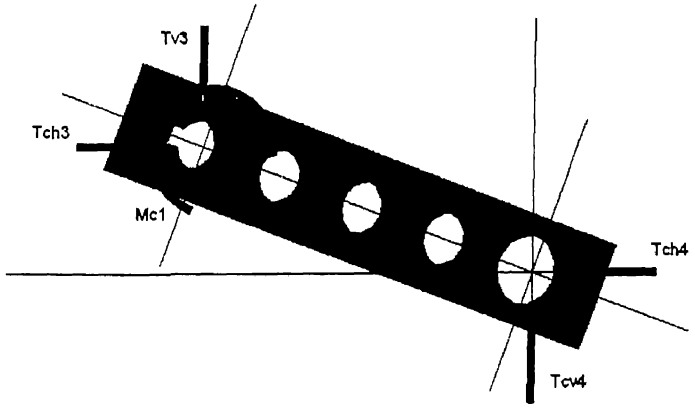


Figure 4.18. Forces and moments acting on four bar link

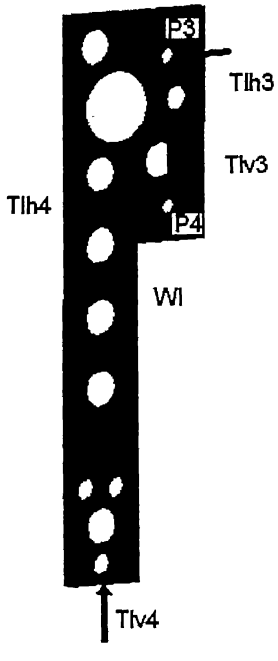


Figure 4.19. Forces and moments acting on the leg on ground

All the forces acting on the leg are as shown in Figure 4.19. All vertical and horizontal forces due to the body and the leg in air which act on the connectors are transferred to the supporting leg. The vertical forces are balanced by the reaction from the ground. The horizontal forces are balanced by the friction between the ground and the foot. The vertical force acting at C1 will create a torque about P4 and since the joint at P4 is free, the body will tend to rotate about P4. The torque applied by the lower hip motor exerts a force perpendicular to the connector at P4. However, since the leg is on the ground, all forces are balanced by the ground reaction force and the ground frictional forces and the leg can not rotate about C1. At the same time the body is free to rotate, as it is supported by thrust bearings. In case the torque is greater than the torque required for overcoming the body weight forces; the body will start rotation about P4. The torque, moment and forces acting on the leg were calculated as given below:

$$
\begin{aligned}
&Tlv3 = -Tbv2 = 116.5N \\
&Tlv4 = -Tlv3 + wl \times 9.81 = -116.5 - 1.8 \times 9.81 = -134N \\
&Tlh3 = -Tbh3 \\
&Tlh4 = -Tbh4 \\
&Ml4 = Mb4 + Tbv2 \times 32.5 = 27044.5Nmm
\end{aligned}
\qquad \text{---------- (4.21)}
$$

Torque required for the lower hip motor was calculated from the following equation.

$$T2 = Tlv3 \times 100 \times \cos(\theta) - Tlh4 \times 100 \times \sin(\theta) \qquad \text{---------- (4.22)}$$

The rotation angle for one step motion varies from 0° to -180°. Differentiating the torque equation (4.15) and equating it to zero, the angle for maximum torque was calculated. Similarly the angle for zero torque was calculated by equating the torque equation to zero. The Maximum value of torque on second hip motor is 12.0585 Nm at Θ=-2.9452rad (-168.7475°). For rotation angle Θ = -1.5168rad (-86.9062) torque will be zero. The torque calculation of hip motor connected to the leg in air for one step is shown in Figure 4.20.

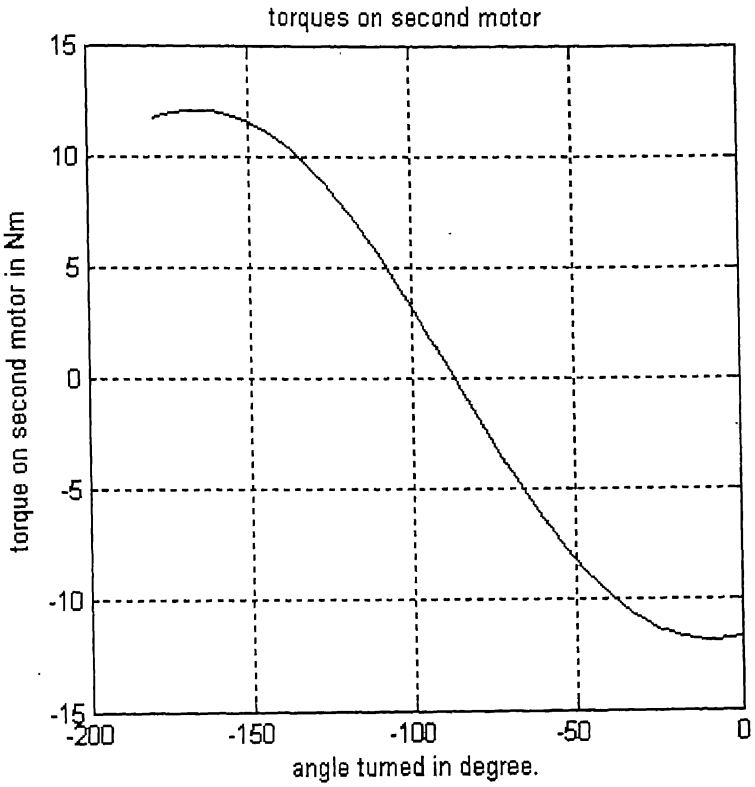


Figure 4.20. Variation of torque acting on second motor.

The maximum bending moment acts on the leg which is on the ground and it was calculated to be 27.044Nm. This leg also caries maximum amount of compressive force (134N). In comparison the bending moment and forces on the leg in air is very small.

All the forces calculated above are for one step when the first leg is in air and the second leg is on the ground. For the next step the first leg will support the biped and the second leg travels in air. Therefore the forces and bending moment are reversed and the first leg caries the maximum load and is subjected to maximum bending moment. Figure 4.21 shows the torque variation of first and second hip motor for two consecutive steps. The tensile forces on the first and second leg for two consecutive steps are shown in Figure 4.22. The bending moments on the first and second leg for two consecutive steps are also shown in Figure 4.23.

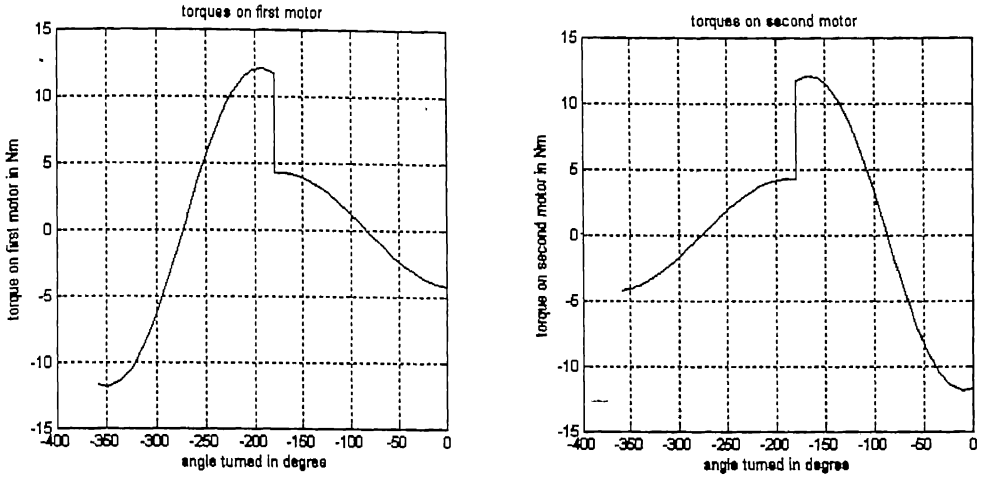


Figure 4.21. Torques acting on hip motors for two consecutive steps.

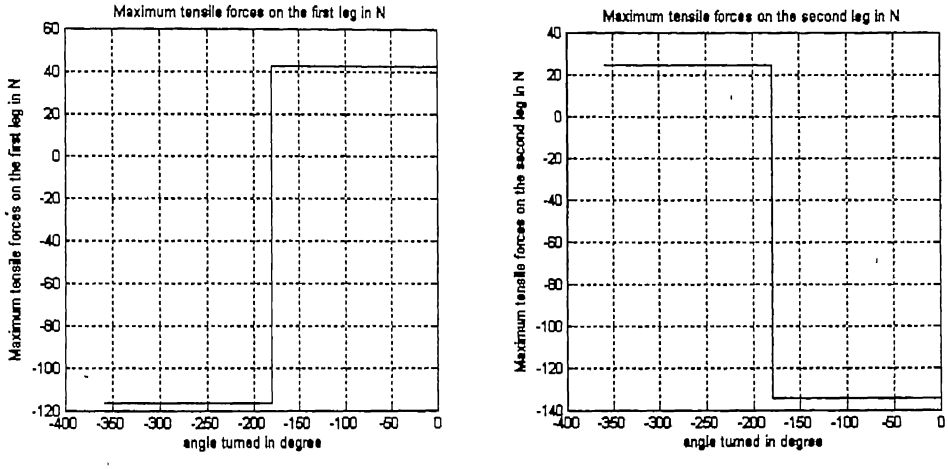


Figure 4.22. The tensile forces acting on the two legs for two consecutive steps.

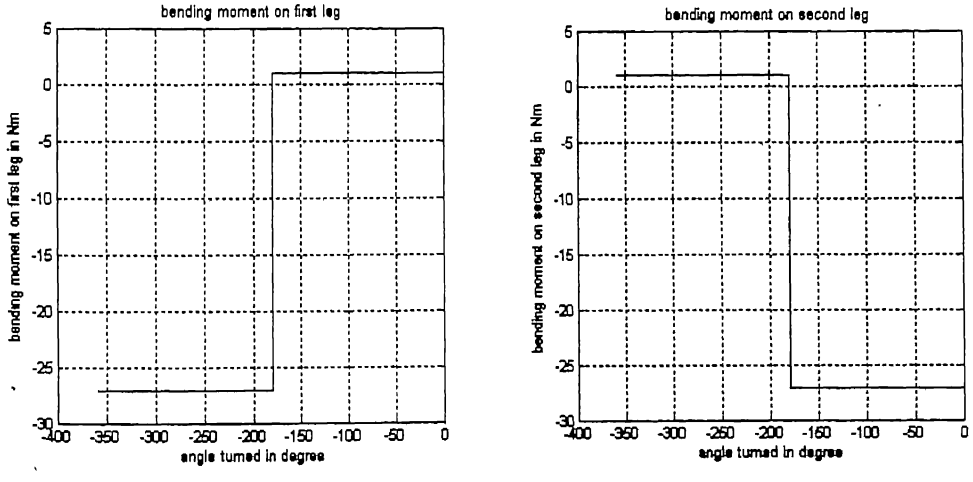


Figure 4.23. The bending moments acting on the two legs for two consecutive steps.

The shear forces on the connecting shaft of first and second leg for two consecutive steps are shown in Figure 4.24.

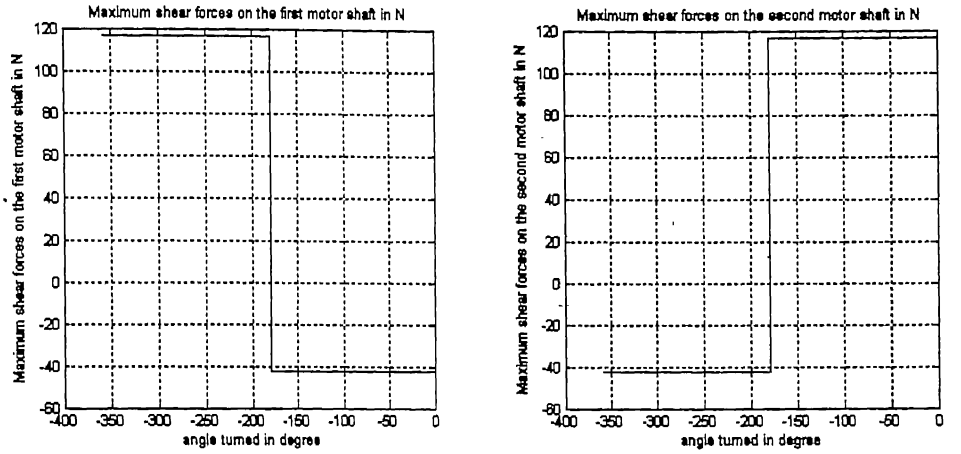


Figure 4.24. The shear forces acting on the shafts of hip motors for two consecutive steps.

4.3 Actuators

The basic element in driving the mechanical design is the choice of actuators. The robot joints are required to supply large forces and torques to move the robot forward. The hip joint has to support a maximum torque of 12.5Nm and if torso load increases, the torque increases up to 15Nm. By using factor of safety of 1.5, the maximum torque becomes 22.5Nm, two Maxon 80 Watt 25Nm DC servomotors with gearbox are used for the hip joints. The gearbox reduces the speed of the four bar links actuating the legs, to about one revolution in 2.5 seconds. The two motors at the ankle are required to support lower torques and two Maxon 40Watt servomotors with inbuilt gearbox have been used. Encoders having 500 counts per revolution have been used for obtaining the position feedback. A DELTA TAU PMAC - 104, 8 axes motor controller is used to control the motors through MAXON 50/5 power amplifiers. Power to the motors is supplied from an onboard 18V lead acid battery.

4.4 Kinematics

Robot kinematics relates the joint angles and the position of the links and foot on the ground. In forward kinematics, for known angles of joints the link positions are

calculated. In inverse kinematics for desired position of the foot (e.g. tracking a trajectory) the required angles can be calculated. In the statically stable biped robot, two feet, two legs and the body will be moving with respect to the foot on the ground. If the angles of the joints are known, the position of all the parts can be calculated by using the forward kinematics. During locomotion, the whole body and one leg (in air) will be actuated with respect to the other leg (on ground) by the hip motor. The other hip motor will only transform the leg in air with respect to the body. The rotation of the ankle motor of the foot on the ground will rotate the whole body about that foot. Whereas the ankle motor of the other foot will rotate that foot only. The detail mathematical analysis for kinematics is given as follows.

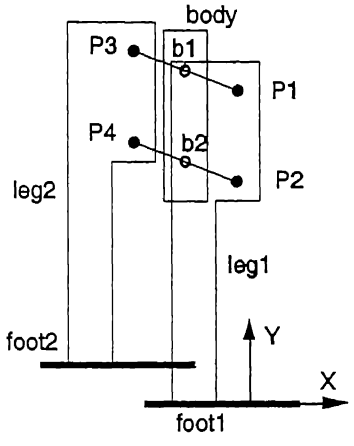


Figure 4.25. Basic frame assignment.

The basic frame assignments are as shown in Figure 4.25. Hip motor at b1 will rotate the whole body and leg in air about P1 = [P1x P1y P1z]. The transformation matrix is calculated below.

$$T1 = \begin{bmatrix} 1 & 0 & 0 & P1x \\ 0 & 1 & 0 & P1y \\ 0 & 0 & 1 & P1z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -P1x \\ 0 & 1 & 0 & -P1y \\ 0 & 0 & 1 & -P1z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{------ (4.23)}$$

$$T1 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & P1x - P1x\cos(\theta) + P1y\sin(\theta) \\ \sin(\theta) & \cos(\theta) & 0 & P1y - P1y\cos(\theta) - P1x\sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation matrix for the rotation about b1 will be

$$T2 = \begin{bmatrix} 1 & 0 & 0 & b1x \\ 0 & 1 & 0 & b1_Y \\ 0 & 0 & 1 & b1z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -b1x \\ 0 & 1 & 0 & -b1_Y \\ 0 & 0 & 1 & -b1z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

------ (4.24)

$$T2 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & b1x - b1x\cos(\theta) - b1_Y\sin(\theta) \\ -\sin(\theta) & \cos(\theta) & 0 & b1_Y - b1_Y\cos(\theta) + b1x\sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore the total transformation matrix will be

$$T = T2 * T1$$

$$T = \begin{bmatrix} 1 & 0 & 0 & C1 \\ 0 & 1 & 0 & C2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

------ (4.25)

Where

$$C1 = \begin{pmatrix} (b1x - b1x\cos(\theta) - b1_Y\sin(\theta)) + (P1x - P1x\cos(\theta) - P1_Y\sin(\theta)) \times \cos(\theta) - \\ (P1_Y - P1_Y\cos(\theta) + P1x\sin(\theta)) \times \sin(\theta) \end{pmatrix}$$ --- (4.26)

$$C2 = \begin{pmatrix} (b1_Y - b1_Y\cos(\theta) + b1x\sin(\theta)) + (P1x - P1x\cos(\theta) - P1_Y\sin(\theta)) \times \sin(\theta) + \\ (P1_Y - P1_Y\cos(\theta) + P1x\sin(\theta)) \times \cos(\theta) \end{pmatrix}$$ --- (4.27)

Since the transformation matrix contains an identity rotation matrix, therefore the total transformation will not consist of any rotation but only translation. The body, leg which is in air and the foot of that leg will be transformed through this matrix. The new position of all of these can be calculated by multiplying this matrix with the old position vectors.

$$b1_n = T \times b1 \quad \& \quad b2_n = T \times b2$$
$$P3_n = T \times P3 \quad \& \quad P4_n = T \times P4$$
$$body_n = T \times body$$
$$leg2_n = T \times leg2$$
$$foot2_n = T \times foot2 \qquad \text{------- (4.28)}$$

Where, the subscript "n" shows the new value of the vector. Since the hip motor at b2 is also simultaneously rotating, the leg in air along with the foot of that leg will translate in a similar manner. The transformation matrix will be

$$T = \begin{bmatrix} 1 & 0 & 0 & C1 \\ 0 & 1 & 0 & C2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{------ (4.29)}$$

Where

$$C1 = \begin{pmatrix} (P3x - P3x\cos(\theta) - P3y\sin(\theta)) + (b1x - b1x\cos(\theta) - b1y\sin(\theta)) \times \cos(\theta) - \\ (b1y - b1y\cos(\theta) + b1x\sin(\theta)) \times \sin(\theta) \end{pmatrix} \qquad \text{--- (4.30)}$$

$$C2 = \begin{pmatrix} (P3y - P3y\cos(\theta) + P3x\sin(\theta)) + (b1x - b1x\cos(\theta) - b1y\sin(\theta)) \times \sin(\theta) + \\ (b1y - b1y\cos(\theta) + b1x\sin(\theta)) \times \cos(\theta) \end{pmatrix} \qquad \text{--- (4.31)}$$

The new position vector of P3, P4 and the new position of the leg and foot can be calculated by multiplying their old values with this matrix.

$$P3_n = T \times P3 \quad \& \quad P4_n = T \times P4$$
$$leg2_n = T \times leg2_n$$
$$foot2_n = T \times foot2_n \qquad \text{----- (4.32)}$$

The new position vectors can be obtained by using these transformations. Now the new position of the whole body is known. These computations are only for walking straight. If the ankle motor of the foot on ground also rotates then the whole body (including the leg of that foot) will rotate about the ankle point. The rotation of the other ankle motor will

only rotate the foot which is in air. The transformation matrix for the rotation about the ankle motor of the foot on the ground is given by

$$T = \begin{bmatrix} 1 & 0 & 0 & f1x \\ 0 & 1 & 0 & f1y \\ 0 & 0 & 1 & f1z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -f1x \\ 0 & 1 & 0 & -f1y \\ 0 & 0 & 1 & -f1z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

------- (4.33)

$$T = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & f1x - f1x\cos(\theta) - f1z\sin(\theta) \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & f1z + f1x\sin(\theta) - f1z\cos(\theta) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$leg1 = T \times leg1_n \quad \& \quad leg2 = T \times leg2_n$$

$$foot2 = T \times foot2_n \quad \& \quad body = T \times body$$

------- (4.34)

The transformation matrix for the rotation about the ankle motor of the foot in air will be

$$T = \begin{bmatrix} 1 & 0 & 0 & f2x \\ 0 & 1 & 0 & f2y \\ 0 & 0 & 1 & f2z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -f2x \\ 0 & 1 & 0 & -f2y \\ 0 & 0 & 1 & -f2z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

------- (4.35)

$$T = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & f2x - f2x\cos(\theta) - f2z\sin(\theta) \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & f2z + f2x\sin(\theta) - f2z\cos(\theta) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The foot of the leg in air will only rotate through this matrix about the ankle of that foot.

$$foot2 = T \times foot2_n$$

------- (4.36)

A MATLAB simulation has been made for checking the forward kinematics. Some results of the simulation are show below in Figure 4.26 and Figure 4.27.

Simulation results for walking in a straight line

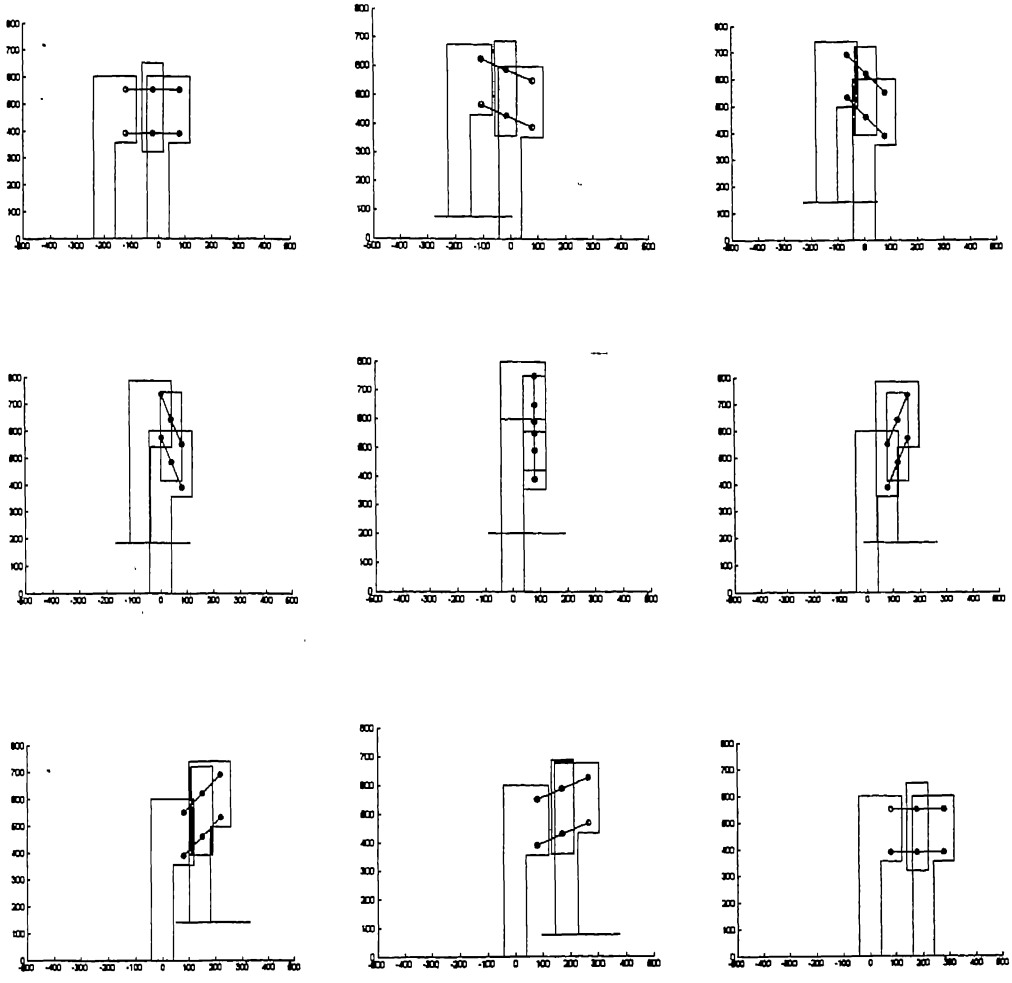


Figure 4.26. Results of forward kinematics for straight walking.

Simulation results for turning at a given angle

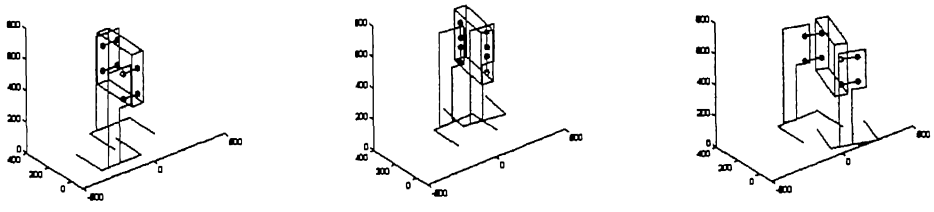


Figure 4.27. Results of forward kinematics for turning.

4.5 Control System

Natural walking of humans consists of two phases: stable and unstable phase. In the case of the statically stable robot it consists of only one phase, the stable phase, which is again divided into the independent control of hip joint and the ankle joint. It has been pointed out in physiological studies that the control of human motion has a hierarchical structure. At the highest level there is the motor cortex, while at the lower level, there exists feedback loops between the muscles and the spinal cord. A hierarchal control network, emulating the human motor control system, having a central controller and a lower order motion controller for the joints was designed. The layout of the control architecture is as shown in Figure 4.28.

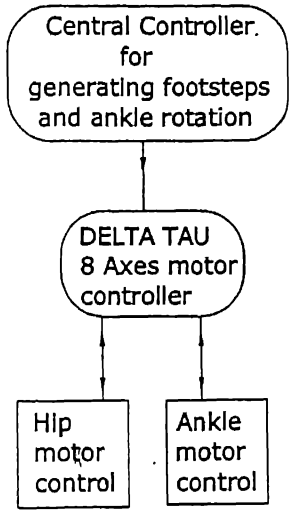


Figure 4.28. The hierarchal control network.

The control algorithms for forward kinematics, trajectory planning etc. are carried out in the central controller which is PC based. After the desired motion is calculated the angles are passed on to the lower level motor controller. The heart of the motor control system is a Motorola DSP, known as Programmable Multi Axes Controller (PMAC) operating at 80 MHz manufactured by Delta Tau. The overall layout of the control system is as shown in Figure 4.29 below. Here the PMAC can either work as a standalone unit or in conjunction with a host computer. As seen in the figure the PMAC connects to the host computer through the serial port. The PMAC in turn is connected to power

amplifiers (MAXON ADS 50/5) as it itself cannot drive the motors. It sends signals to the amplifiers which in turn amplify them and send them to the motor to be controlled. The feedback from the encoders comes back straight to the PMAC where it is used to obtain precise positioning of the motor shaft.

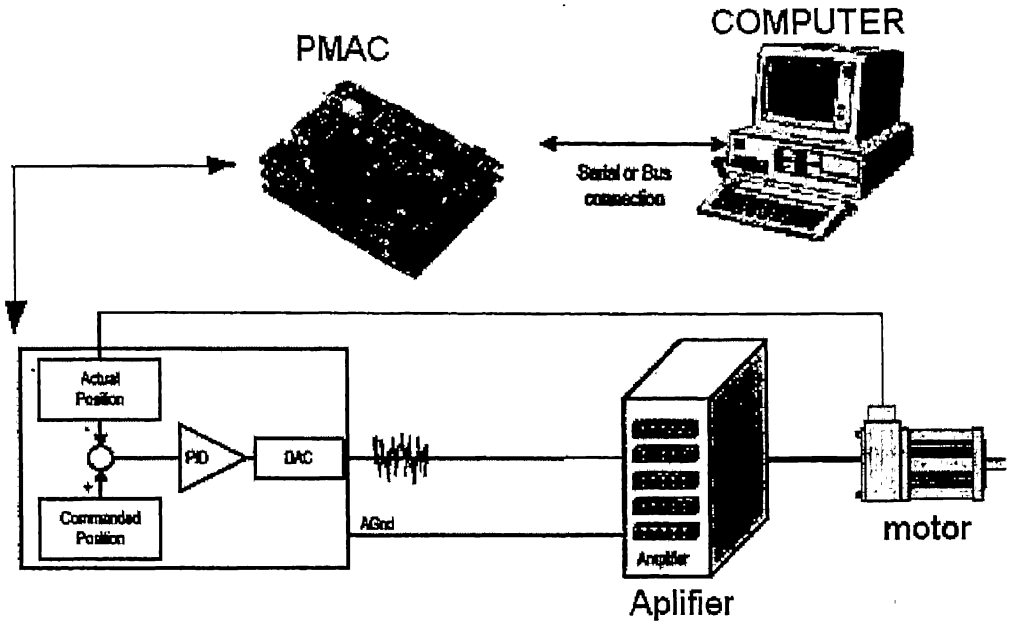


Figure 4.29. Layout of the controller.

The PMAC2 PC/104 used as the motor controller is suitable for small applications like this and it can control up to 8 motors simultaneously. The basic specifications of the PMAC are given below.

- 40/60/80 MHz DSP56311 CPU
- 128k x 24 SRAM user memories
- 512k x8 flash memory for user backup and firmware
- 8-bit parallel PC/104 host computer interface
- 4 channels axis-interface circuitry, each including:
    - -12-bit +/-10V differential analog output (filtered PWM)
    - -3-channel differential single-ended encoder input
    - -Pulse-and-direction output pair

There are two ways in which PMAC can be interfaced to control the motors. First is through the executive program provided by Delta Tau i.e. PEWIN. The other method is

to write routines in 'C' language and then communicate with the PMAC. For controlling through 'C' routines with PMAC, few standard routines have been provided in C language. First and the most important part in using PMAC is the configuration of PMAC. Before using the PMAC it has to be set up to suit the application. The most important parameters to be set up are the I-variables, P-variables, Q-variables and M-variables. All of these are described briefly below.

I-Variables (initialization or setup variables) determine the personality of the card for a given application. They are at fixed locations in memory and have pre-defined meanings. Most are integer values, and their range varies depending on the particular variable. There are 1024 I-variables, from I0 to I1023, and they are organized as follows:

I0 -- I79: General card setup

I100 -- I184: Motor #1 setup

I185 -- I199: Coordinate System 1 setup

I900 -- I979: Encoder 1 - 16 setup

P-variables are general purpose variables. They are 48-bit floating-point variables at fixed locations in PMAC's memory, but with no pre-defined use. There are 1024 P-variables, from P0 to P1023. P-variables can be used in programs for any purpose desired: positions, distances, velocities, times, modes, angles, intermediate calculations, etc. A given P-variable means the same thing from any context within the card; all coordinate systems have access to all P- variables (contrast Q-variables, which are coupled to a given coordinate system, below). This allows for useful information passing between different coordinate systems.

Q-variables, like P-variables, are general-purpose user variables: 48-bit floating-point variables at fixed locations in memory, with no pre-defined use. However, the meaning of a given Q-variable (and hence the value contained in it) is dependent on which coordinate system is utilizing it. This allows several coordinate systems to use the same program (for instance, containing the line X (Q1+25) Y (Q2), but to do have different values in their own Q variables (which in this case, means different destination points).

M-variables: In order to permit easy user access to PMAC's memory and I/O space, M-variables are provided. The concept of m variables is quite similar to that of pointers in C language. While interacting with the PMAC, it might be necessary to know the current values of encoders, motor angle position etc. These can be obtained form specific memory locations pointed to by M variables. Generally, a definition only needs to be made once, with an on-line command. On PMAC with battery backup, the definition is held automatically. On PMAC with flash backup, the **SAVE** command must be used to retain the definition through a power-down or reset. The user defines an M- variable by assigning it to a location, and defining the size and format of the value in this location. There are 1024 M- variables (M0 to M1023), and as with other variable types, the number of the M-variable may be specified with either a constant or an expression: M576 or M (P1+20) when read from; the number must be specified by a constant when written to. The definition of an M-variable is done using the "defines-arrow" (->) composed of the minus-sign and greater-than symbols.

There are mainly two types of commands for working with PMAC namely **Online Commands** and **Program Commands.** Online Commands are the commands that can be straight away given at the command prompt in PEWIN, where as the program commands are the commands that are written either into a **Motion Program** or **PLC Program.** Program commands are executed only when the program is executed. Some of the important online as well as program commands are given here.

- ***<Control-A>*** this command aborts all motion programs and stops all non-program moves on the card.
- ***<#{constant}>*** this command makes the motor specified by **{constant}** the addressed motor (the one on which online motor commands will act).
- ***<&{constant}>*** this command makes the coordinate system specified by **{constant}** the addressed coordinate system (the one on which on-line coordinate-system commands will act).
- ***<J+ J- J/ J:{constant}>*** J+ causes the addressed motor to jog in the positive direction indefinitely. Similarly J- causes the motion in negative direction, J/ causes

the motion to stop and J:{constant} is used to move the motor through a fixed encoder counts.

- *<I{constant}>* this command causes PMAC to report the current value of the specified I-variable or range of I-variables.
- *<SAVE>* this command causes PMAC to copy setup information from active memory to non-volatile memory, so this information can be retained through power-down or reset. Its exact operation depends on the type of PMAC used.
- *<{axis}{data}[{axis}{data}...]>* - This is the basic PMAC move specification statement. It consists of one or more groupings of an axis label and its associated value. The value for an axis is scaled (units determined by the axis definition statement); it represents a position if the axis is in absolute (ABS) mode or a distance if the axis is in incremental (INC) mode. The order in which the axes are specified does not matter. This command tells the axes *where* to move; it does not tell them *how* to move there. Other program commands and parameters define how; these must be set up ahead of time.
- *ABS* - The **ABS** command without arguments causes all subsequent positions in motion commands for all axes in the coordinate system running the motion program to be treated as absolute positions. This is known as absolute mode, and it is the power-on default condition.
- *DELAY{data}* - This command causes PMAC to keep the command positions of all axes in the coordinate system constant (no movement) for the time specified in **{data}**. There are three differences between **DELAY** and **DWELL**. First, if **DELAY** comes after a blended move, the **TA** deceleration time from the move occurs within the **DELAY** time, not before it. Second, the actual time for **DELAY** does varies with a changing time base (current % value, from whatever source), whereas **DWELL** always uses the fixed time base (%100). Third, PMAC precomputes upcoming moves (and the lines preceding them) during a **DELAY**, but it does not do so during a **DWELL**. A **DELAY** command is equivalent to a zero-distance move of the time specified in milliseconds. As for a move, if the specified **DELAY** time is less than the acceleration time currently in force (**TA** or **2*TS**), the delay will be for the acceleration time, not the specified **DELAY** time.

- ***ELSE ENDIF IF*** - This command allows conditional branching in the program. With an action statement or statements following on that line, it will execute those statements provided the condition is true (this syntax is valid in motion programs only). If the condition is false, it will not execute those statements; it will only execute any statements on a false condition if the line immediately following begins with ELSE. If the next line does not begin with ELSE, there is an implied ENDIF at the end of the line.
- ***WHILE ({condition}) {action}*** - This statement allows repeated execution of a statement or series of statements as long as the condition is true. It is the PMAC only looping construct.

The development of the control system was a step-by-step process. First the connection of the motors, encoders and the PMAC was checked to make sure that everything was properly connected. Initially only the hip motors were connected to obtain the straight motion of the robot without any turning incorporated. Initially when motors are connected they need to be setup for PMAC to control them. Using PEWIN the corresponding I-variables for the motors are updated to get the desired response of the motors. For making sure that the encoder connections are right the position of the motors need to be checked through PEWIN, if the encoder counts change with the rotation of the motors the connections are proper. Now the motors are checked with simple jog commands to see if they are working properly.

The next step was tuning of motors which was done using the PID tuning utility in PEWIN. The tuned parameters were then manually changed, depending on the step response of the system. The motors operate in a closed loop mode, when a particular position is commanded and the actual position attained by the motor is fed back through the encoders on the motors. What the PID controller does is depending on the error encountered in the position of the motor the commanded position is changed through a closed loop so that the error is minimized. Depending on the values of these parameters the input is changed as per the error present. Two other loops which control the time of motion, are present i.e. velocity feed forward and acceleration loops. The process of tuning is basically finding the right values for these gains so that the response produced is

nearly same as the commanded position. The result obtained after tuning of the motors which depict the commanded as well as the followed position are shown below in Figure 4.30. The figure also shows the gain values used to get the shown response.

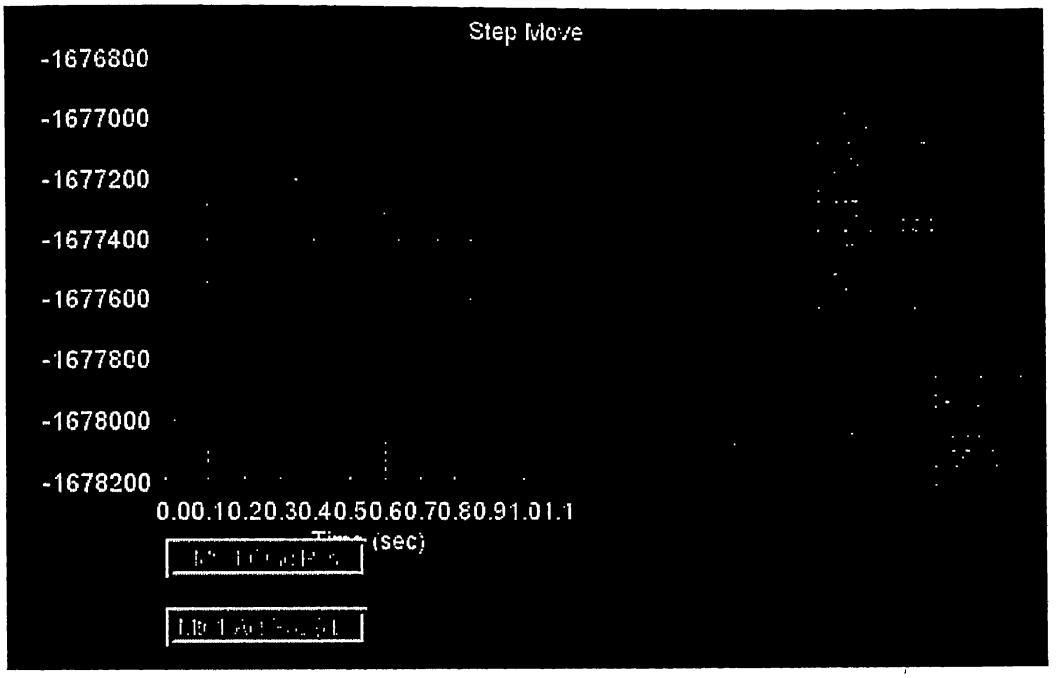


Figure 4.30. Step response of the motor.

After the tuning of the motors was done, the motors were run directly through the jog commands. Next a simple program was developed which could run the motors though a 'C' program. A simple program written in PWIN software is given below:

OPEN PROG 1 CLEAR
LINEAR
TA5
TS100
TM2000
INC
X0 Y0
X0 Y0
ABS
WHILE (P1>0)
X(M70) Y(M71)

```
ENDWHILE
CLOSE
```

In the Program the first line opens a buffer for writing the program, or to clear the buffer if it already had anything in it. After that it specifies linear mode of operation wherein all the motors in the co-ordinate systems move in equal proportion to each other. TA, TS and TM specify the time of acceleration, time of S-Curve and time of Move respectively. All the units here are in milliseconds. INC specifies the incremental mode and then two zero moves are made to move the motors to home positions if they are slightly off. Then again the mode is set to absolute mode with ABS command. The main point here to note is that in absolute mode the motors move only if there is any change in the given variable, else they do not move. This is the intended function as we want the motors to move only when we update the value of the variable. The next few lines specify moves for the X and Y axes as per the value of M70 and M71 respectively as long as P1>0. The structure of this program is such that it keeps on waiting in the loop for values of M70 and M71 to change, and as soon as these values change the motors are moved by the corresponding amount. When operating through PEWIN first the program is executed then the values are changed to get the desired motion in the motors. While operating through the 'C' program the values are updated by the program as per the input given by the user. The end of the program specifies a command to close the buffer. This program is for controlling only two motors whereas for four motors the program specifies two more axes U and V. These moves can also be obtained by a simple C-program. What the 'C' program does is first check to see if PMAC is present. Then it simply sends commands to PMAC for the desired motion. An application was developed that takes from the user as input the leg on which the step is supposed to be taken along with the angle to be turned. It then executes the required step.

Chapter 5

Trajectory planning and obstacle avoidance

The main objective of this chapter is first to calculate a smooth path from the initial point to the final goal point avoiding all the obstacles in the work space and then to calculate the angles required for each ankle motor to follow the desired path.

5.1 Trajectory planning

The statically stable biped has four DOF, two for moving forward and two for turning. The procedure for following a trajectory is to position the centre of gravity of the biped on the trajectory as it moves. The trajectory is given in the form of finite number of control points on the desired trajectory. If few control points are given then the trajectory is made by making an interpolated curve through these points. The interpolated curve, used for the calculation is a Hermite curve (third order interpolation).

The rotation of the ankle motor of the foot on the ground rotates the whole torso and the leg in air, where as the rotation of the ankle motor of the leg in air, rotates only the foot of that leg. Therefore the ankle motor of the foot on the ground will mainly contribute for the rotation of the biped whereas the rotation of the other ankle motor will only make the aerial foot parallel to the trajectory. The required angles of the ankle motors are calculated for every 180° rotation of hip motors. In order to follow a trajectory it is required to rotate both the ankle motors.

5.1.1 Calculation of desired trajectory from given control points

The equation of a third order interpolation hermite curve can be expressed as

$$P(t) = \sum_{i=0}^{3} C_i \times t^i \quad ; \qquad \text{where } 0 \le t \le 1 \qquad \text{------- (5.1)}$$

$$P(t) = C_3 \times t^3 + C_2 \times t^2 + C_1 \times t^1 + C_0 \times t^0 \qquad \text{------- (5.2)}$$

Where C_i is a vector. The above equation can be written as

$$P(t) = \underline{T}^T C \qquad \text{------- (5.3)}$$

Where P(t) is the coordinate of a point at time "t".

$$T = [t^3 \quad t^2 \quad t \quad 1]^T$$
$$C = [C_3 \; C_2 \; C_1 \; C_0]^T \qquad \text{------- (5.4)}$$

In order to find the four unknowns four conditions are required. Taking the first and second point coordinates (P_0 and P_1) and the tangent vectors at those points (P_0' and P_1') the above equation becomes,

$$P(t) = T^T M_H V \qquad \text{------- (5.5)}$$

Here M_H is the hermite matrix.

$$M_H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$V = [P_0 \quad P_1 \quad P'_0 \quad P'_1]^T \qquad \text{------- (5.6)}$$

This is the curve between two points, curves through more than 2 points can also be interpolated but then C^2 continuity is required.

5.1.2 Calculation of ankle angles to follow desired trajectory

Due to the physical constraints, the robot can turn left/right by 10° and right/left by 35° in one step without foot interference. The Figure 5.1 shows the two constraints on the angle for turning.

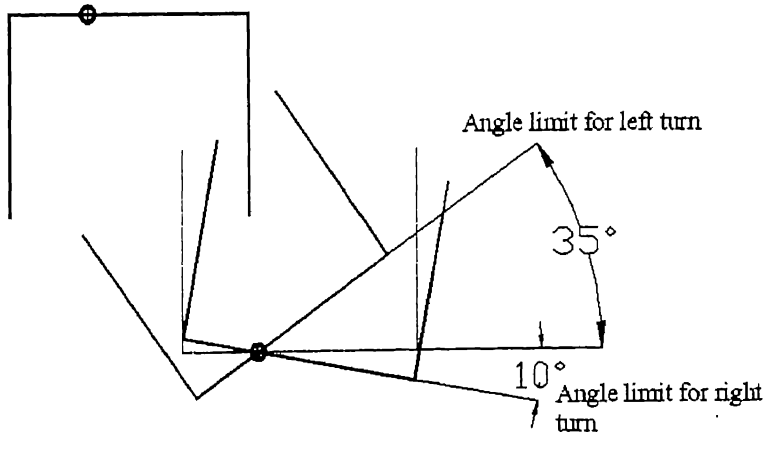
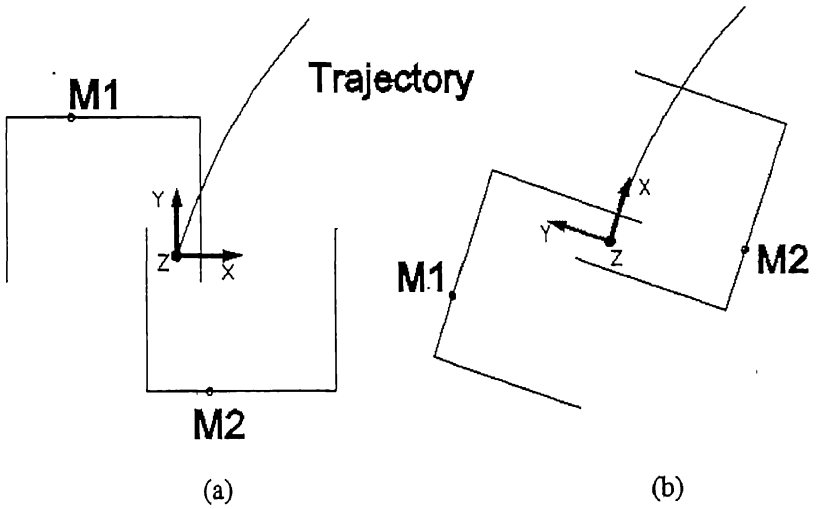


Figure 5.1. Limits of angles of ankle motor.

Since there is a limitation in rotation of the ankle motor, it is best to keep the biped tangential to the trajectory at the starting point. The biped can come to this configuration from its initial configuration either by turning or back stepping and turning. In the simulation, the x-axis of the local coordinate system (attached to the foot) is in the direction of walking as shown in Figure 5.2 (a). First the biped is assumed to be at the starting point parallel to the x axis of the base frame, then by simply turning or back stepping it turns through an angle such that the x-axis becomes parallel to the tangent at the starting point, as shown in Figure 5.2 (b).



(a) (b)

Figure 5.2. Desired initial orientation of robot.

If the starting point of the trajectory is P1= [$P1_X$ $P1_y$] and the robot is positioned such that the CG of the biped falls on P1, then the position coordinates of the motors of two ankles (M1 and M2) before rotation is given by (Figure 5.2(a))

$$M1 = [P1_X - 150 \quad P1_Y + 184 \quad 0]^T$$
$$M2 = [P1_X + 50 \quad P1_Y - 184 \quad 0]^T \qquad \text{------- (5.7)}$$

If the angle of the tangent at starting point is Θ then,

$$[M1 \quad M2]_{initial} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} [M1 \quad M2] \qquad \text{------- (5.8)}$$

The new local coordinate system will be such that the x-axis will be in the direction of the tangent to the curve and y-axis will always be perpendicular to the tangent, z-axis is always vertical. After the robot is positioned at the initial position, the first step is analyzed for which M1 changes and M2 remains same. M1 rotates in air and travels 200mm in the x direction as shown in Figure 5.3

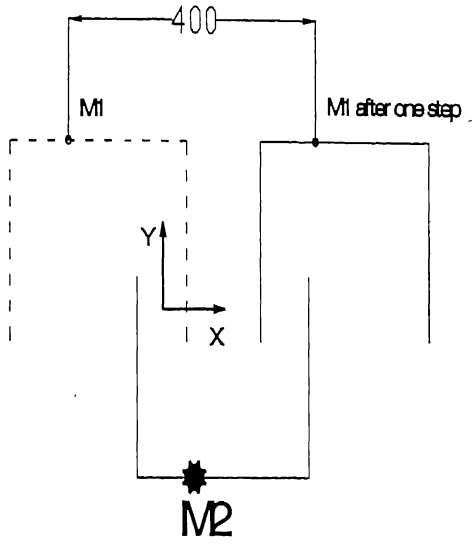


Figure 5.3. Position of M1 after one straight step.

Since the distance between M1 and M2 in y direction is 368mm and the y-axis is perpendicular to the tangent to the trajectory at starting point, the distance between M1 and M2 perpendicular to the trajectory is 368mm. In order to bring the centre of gravity on the trajectory, both M1 and M2 will have to be at the same distance from the trajectory in y-direction. Hence the distance of M1 and M2 from the desired trajectory will be 184mm. After one step the minimum distance of M1 from the desired trajectory will either increase or decrease. In order to make the distance again 184mm, motor at M2 rotates so that the whole body as well as the other leg rotates about M2. The angle of the motor rotation at M2 must be such that the minimum distance of M1 from the desired trajectory becomes 184mm. The point on the trajectory having the minimum distance from M1 is also calculated (P_N) as shown in Figure 5.4.

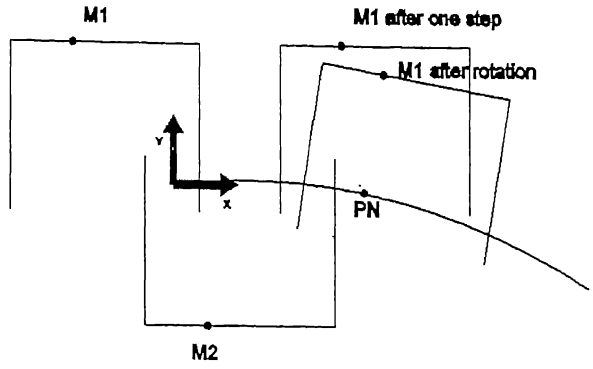


Figure 5.4 Rotation of foot in air about M2

After rotation of the motor at M2, motor at M1 is also rotated to make the second foot parallel to the tangent to the trajectory (Figure 5.5).

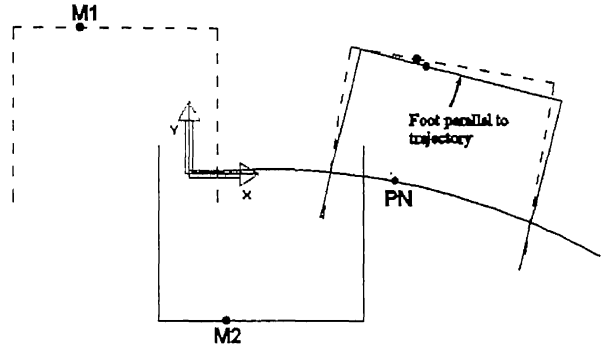


Figure 5.5 Rotation of foot in air about M2

These two rotations are given to the motor at M1 and M2 simultaneously and the rotations are completed before the hip motor complete 180° rotation. The coordinate system will also rotate as the motor at M2 will rotate. The new x-axis will be in the direction of the tangent to the trajectory at P_N and y-axis will be perpendicular to that tangent. For next step M1 will be on the ground and M2 will swing through 400mm in the new x-direction.

Due to the limitation in rotation, if the required angle of rotation is more than the limits, the trajectory can not be followed in a single step approach. The trajectories having high curvature can be followed by back stepping where several front and back half steps are taken to take a big turn standing at the same point. Therefore in the algorithm given above, if the calculated angle required for turning is greater than the

limits then back stepping is used. The flow chart for following a trajectory is as shown in Figure 5.6.

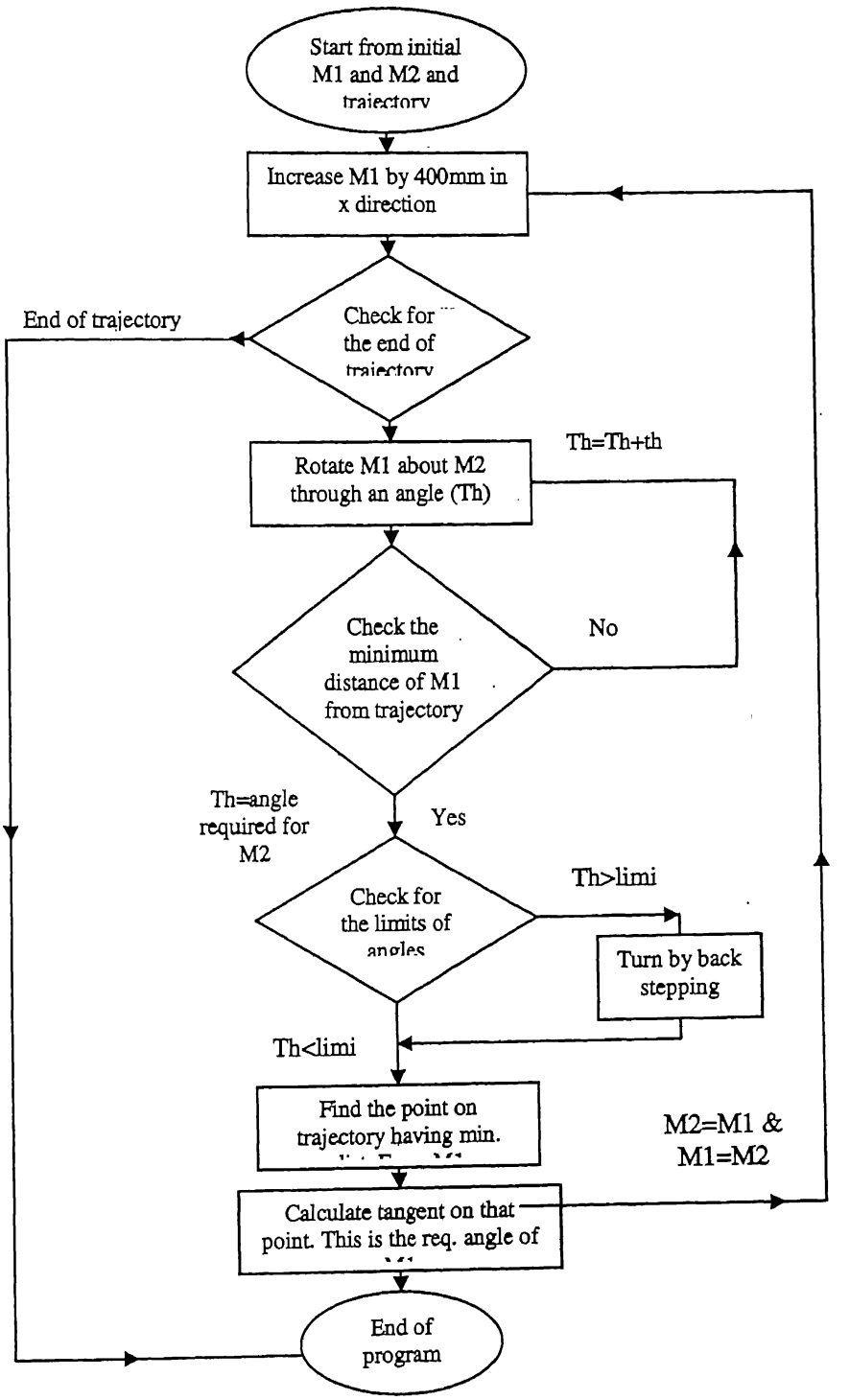


Figure 5.6. Flow chart of the trajectory following process.

5.1.3 Simulation for trajectory following

A MATLAB simulation has been developed for testing the procedure for trajectory following. The inputs are the control points on the trajectory that the robot is supposed to traverse. A sub function makes the trajectory through the points by interpolating the control points by a hermite curve. The main function then calculates the angles required for each motor and then stores the output in text files which later can be used for controlling the motors. The outputs of the simulation are

a) Points on the trajectory.

b) Initial angle of the biped.

c) Angles required for motor at M1 when M1 is in air/ or on the ground.

d) The angles required for motor at M2 when M2 is in air/ or on the ground.

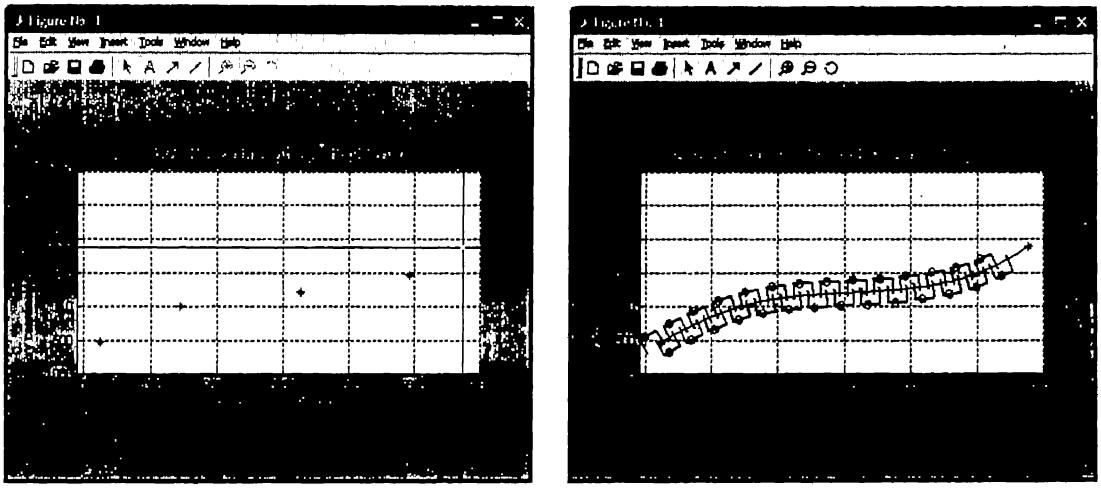
Other than this outputs, a graphical output is also generated, in which the trajectory and foot pattern is displayed.

A MATLAB graphical user interface has also been developed, which has three button inputs, one edit text box and one axes where figures are drawn. In the edit text box the number of control points desired on the trajectory is given. The default value has been given as 5. The first button asks for the points to be traversed and makes the trajectory. The second button calculates the angle pattern for following the trajectory, and the last button is for viewing the animated walking pattern.

Another MATLAB simulation has been made which calculates several paths between two points and then classifies them as simple to follow (no back stepping), difficult (one back stepping) and very difficult (more than one back stepping). All the trajectories which can be followed by single step walking and turning are shown as simple trajectories, and all the trajectories which need the back stepping are shown as difficult trajectories. The trajectories which need more than two back stepping are shown as most difficult trajectories to be followed.

5.1.4 Simulation results for trajectory following

Figure 5.7 (a) Shows the input control points of trajectory and the second Figure 5.7 (b) shows the walking pattern on the trajectory.



(a) (b)

Figure 5.7 Results of trajectory following

Set of angles of the motor at M1 when M1 is on the ground =

[-0.03 -0.04 -0.04 -0.05 -0.03 -0.03 -0.01 -0.00 0.03 0.03 0.04 0.05 0.06]

Set of angles of the motor at M2 when M2 is on the ground=

[-0.03 -0.03 -0.03 -0.05 -0.06 -0.03 -0.02 -0.00 0.01 0.02 0.03 0.05 0.07]

Set of angles of the motor at M1 when M1 is in air=

[-0.05 -0.06 -0.07 -0.10 -0.10 -0.06 -0.04 -0.01 0.02 0.04 0.06 0.09 0.13]

Set of angles of the motor at M2 when M2 is in air=

[-0.08 -0.06 -0.08 -0.11 -0.08 -0.05 -0.02 0.01 0.03 0.05 0.07 0.11 0.12]

Initial theta of biped= [-0.53]

The simulation results for the trajectory following of some other trajectories are given in Figure 5.8.

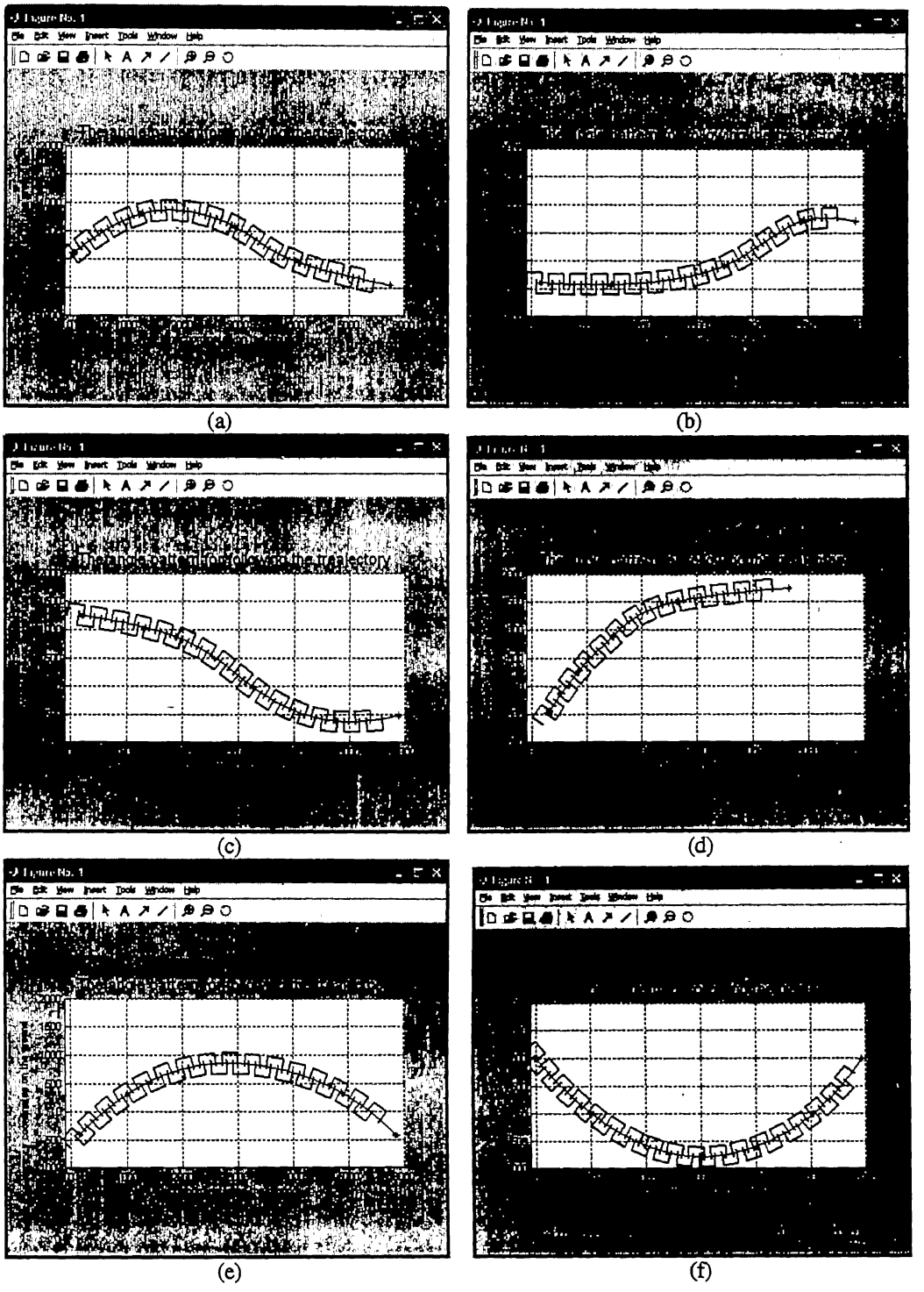


Figure 5.8. (a) to (f) Results of trajectory following.

Figure 5.10 shows the results of the simulation for different difficulty levels of the trajectories between two points. In this figure thin plotted trajectories are the trajectories which can be followed without any back stepping. The thick trajectories can be followed by single back steps, while the thick trajectories are most difficult trajectories to be followed, and need more than one back stepping.

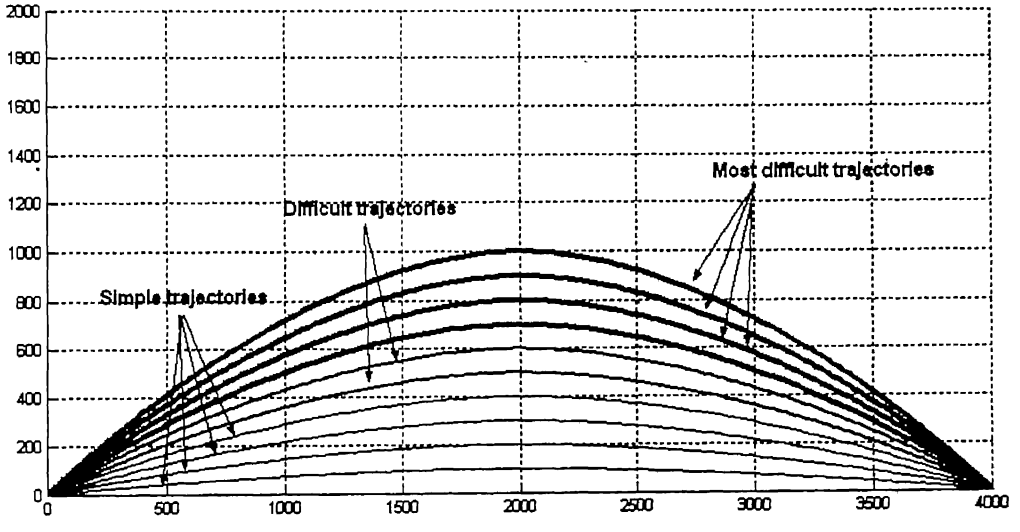


Figure 5.10. Difficulty levels of different trajectories.

5.2 Obstacle avoidance

For real time navigation it is required that the robot should be able to navigate through obstacles in the workspace. The main objective of this section is to compute a feasible path in a given workspace (with only polygonal obstacles) from a given initial configuration to a final goal configuration, avoiding all the obstacles. The robot and all obstacles are taken as two dimensional polygonal objects. The dimensions of the robot for path planning are considered by converting the workspace in C-space (configuration space). All the polygonal obstacles are enlarged in all the directions through a distance equal to the maximum dimensions of the robot in the horizontal plane. The obstacles after enlargement are called C-obstacles and the boundaries of the workspace are shrunk through the same amount. In this C-space the path is planned by taking the robot as a point. There are several different path planning approaches, but unfortunately no general method exists which can work efficiently for all the problems. The two main path planning approaches used here are (i) roadmap method and (ii) potential method and they are discussed below.

(i) ***Roadmap method:*** - In the roadmap method, a standardized path network, called roadmap is generated in C-free space (subspace of the configuration space) having no obstacles, (C-free space = C-space – C- obstacles) and closure-C-free (edges of the obstacles) and then from the initial point to the goal point a path is found in this network only. Various different methods have been proposed for the generation of the roadmap. Such as visibility graph method, voronoi diagram method, free way net method and silhouette method.

Visibility graph method has been used for path planning of the statically stable biped robot. In this approach, the network is generated such that the nodes of the network will be the initial point, goal point and the nodes of the obstacles. All the links of the roadmap are straight lines connecting two nodes such that the complete line lies in the C-free region and Cl-free region, i.e. the line does not intersect the interior of the obstacles. The biggest disadvantage of this method is that it does not generate a path in C-free region instead it generates the path in C-free and Cl-free region, i.e. the robot goes by touching the obstacles. Another disadvantage of this method is the sudden change in the

direction of motion at each node. The advantage of this method is its simplicity i.e. it is less computationally complex, because it only deals with the nodes of the obstacles, and once the whole network is generated for a given work space, then for different paths, the same network is used.

(ii) ***Potential field method:*** - In the potential field method of path planning the robot is considered as a particle starting from the initial position, moving under the influence of several artificial potentials produced by the obstacles and the goal point. The obstacle will produce a repulsive potential whereas the goal point will produce an attractive potential, i.e. the obstacles will try to push the robot away from it whereas the goal point will try to pull the robot towards the goal point. At every point the direction of the resultant force (due to the attractive and repulsive forces) will be in the direction of motion. The biggest disadvantage of this approach is that it cannot always assure a solution, even in cases in which a solution exists. This is because this approach is based on the steepest descent optimization method and hence it can get stuck on the local minimas in the work space.

5.2.1 Path planning of the biped robot

A combined method has been used for planning the path in a C-polygonal workspace for the statically stable biped robot. In this method both the roadmap methods are used for finding out a path from the initial point to the goal point. As discussed earlier, this path will be semi-free path (touching the edges) and not smooth. Therefore potential field method has also been used for smoothing the path. The path planning approach was as follows:

a) Capture the workspace from the top using a vision camera, and take the initial and final configuration of the robot.
b) For the detected obstacles, calculate the C-obstacles and the C-space.
c) Make a roadmap "R" for given initial point, final point and the nodes of C-obstacles.
d) Find a path of minimum length form the initial to the goal point in "R".

e) Apply an artificial negative potential to the obstacles and an attractive potential to the goal point.

f) Smooth the path generated by the roadmap method by potential field method.

Each of these steps is discussed in detail as follows: All the C-obstacles are calculated for the corresponding obstacles and the robot, i.e. they are enlarged by an amount equal to the maximum dimension of the robot. The robot is translated and rotated around an obstacle and all points are found out for which no configuration of the robot is possible which can avoid colliding with the obstacles. All those points are the nodes of the C-obstacle. As a result the nodes of the obstacles get converted into arcs and it is very difficult and computationally complex to use the arcs in the roadmap method. Therefore for reducing the complexity of the problem the obstacle is always kept as an enlarged polygonal obstacle. The following steps were followed for enlarging the obstacles.

a) From the first node draw a line as an outer normal to the edge connecting that node to the next node. (Figure 5.11 (b))

b) On this normal take a point 'P' at a distance "d" from the node. Where d is the maximum span of the robot in the horizontal plane.

c) From point P draw a line parallel to the edge connecting the first node to the next node. (Figure 5.11 (c))

d) Repeat step a-c for all the nodes of the obstacles. (Figure 5.11 (d))

e) Calculate the intersection points of all the lines drawn from the extended points parallel to the edges. (Figure 5.11 (e))

f) Connecting these intersecting points, the C-obstacle is generated. (Figure 5.11 (f))

g) Repeat steps a-f for all the obstacles. The boundaries are also converted into configuration boundaries by the same method, but in this case the normal is taken in the inside direction.

h) After calculating all the C-obstacles and C-boundaries, the configuration space (C-space) is calculated. (Figure 5.12)

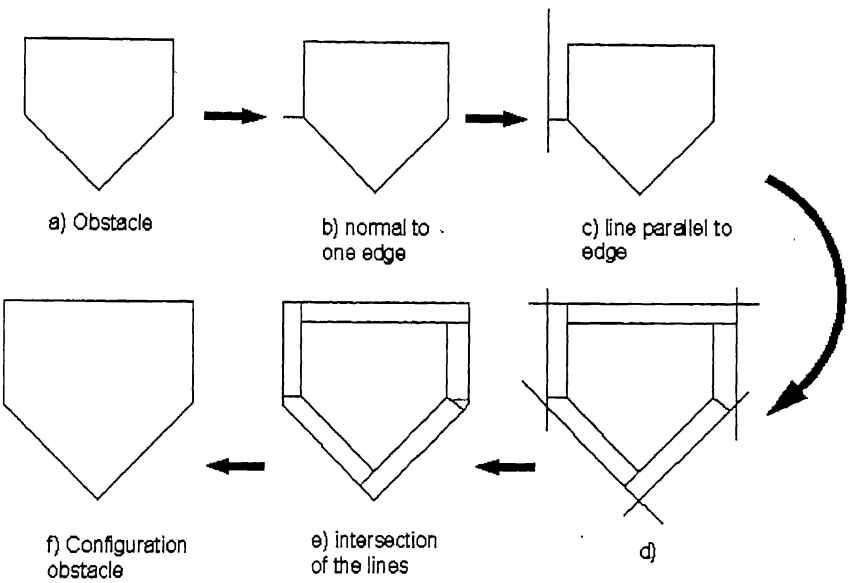


Figure 5.11. Converting obstacle into C-obstacle.

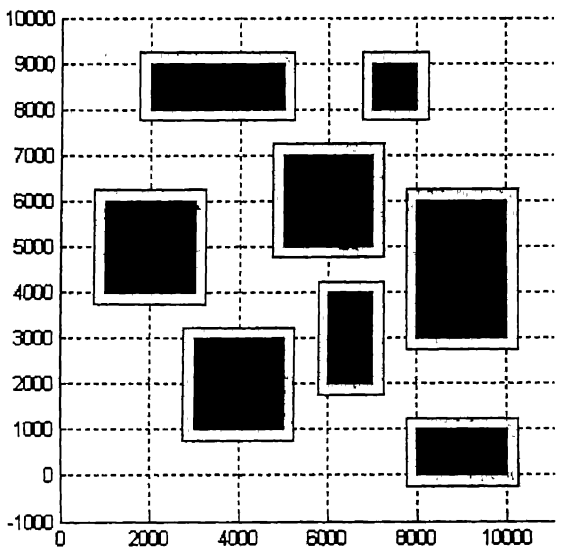


Figure 5.12. C-space.

After making a C-space, the initial and final configurations are entered and then a roadmap is generated.

In the roadmap method, a network is generated and then the path from the initial to final point is searched in this network. Following procedure is followed for generating the network.

a) From the first node of the first obstacle, connect first node of second obstacle by a line.

b) Divide this line into finite no. of points (say 100).

c) For first point of the line, check all the obstacles and find out that the point is inside any obstacle or not.

d) If the point is inside any obstacle, then that line will not be a link of the network, go to step no. g. If the point does not lie inside any obstacle, check for the next point.

e) Repeat step no. c-e for all the points on the line.

f) If all the points of the line are outside the obstacles, the line is included in the network as a link.

g) For all the nodes of the first obstacle, repeat step no. a-f.

h) Repeat a-g for all the obstacles.

i) From steps a-h a network has been generated. In this network add all the edges of all obstacles and the final network is the roadmap.

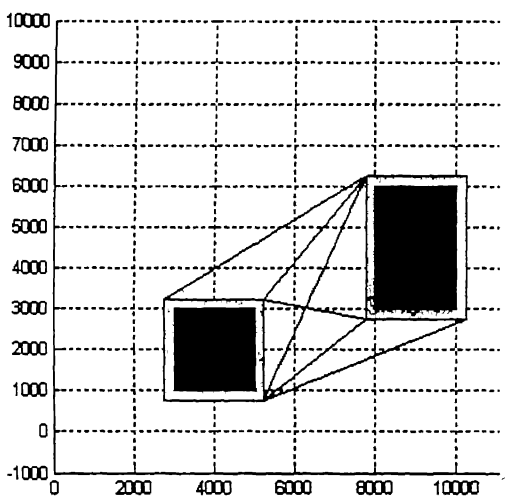
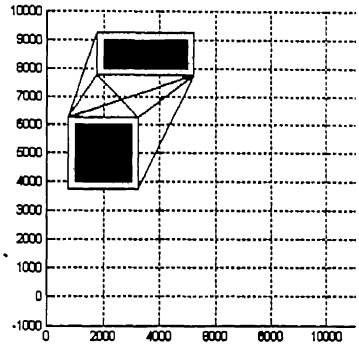
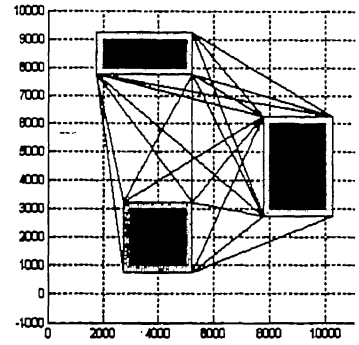


Figure 5.13. Roadmap for two obstacles.

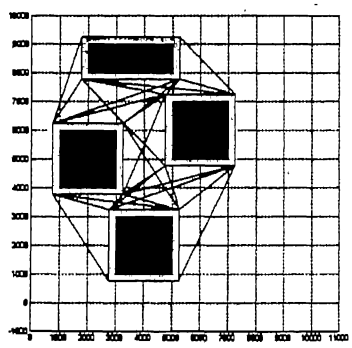
One simulation has been developed for generating a roadmap. The obstacles are given as their nodes and then C-space is generated then roadmap is generated. Figure 5.13 shows the roadmap for two obstacles. The results of the simulation are shown in the Figure 5.14 are for two, three to eight obstacles respectively.



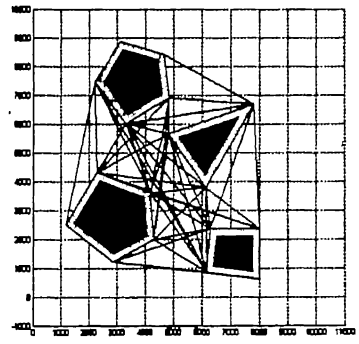
a) 2 obstacles



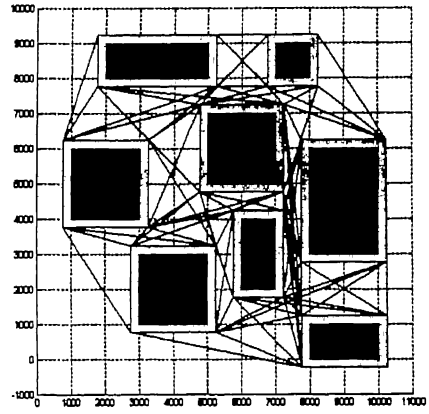
b) 3 obstacles



c) 4 obstacles



d) 4 obstacles



d) 8 obstacles

Figure 5.14. Results of roadmap generation.

The algorithm for checking if a point is inside a polygon is as follows:

a) Draw a vector from the point (P) to the first node of the obstacle (n1), name the vector 'n1p'. (Figure 5.15)
b) Draw a vector from n1 to n2, name this vector n1n2.
c) Find the cross product of the two vectors n1p and n1n2; name the vector 'a1'.

$$\overline{a1} = \overline{n1p} \times \overline{n1n2}$$

d) Calculate the unit vector 'a11' of 'a1'.
e) Repeat step a-d for all the nodes and calculate all the unit vectors.
f) If all the unit vectors are same, the point will be inside the polygon; otherwise it will be outside the polygon.

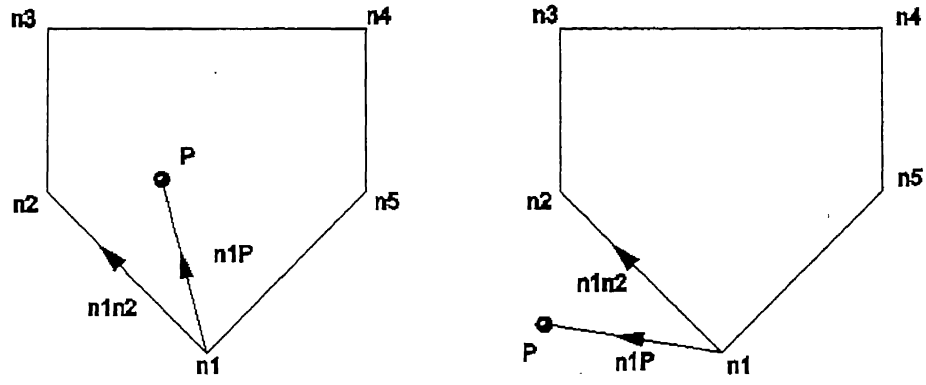


Figure 5.15. The algorithm for checking if a point is inside a polygon.

After generation of the roadmap (network), the path of the minimum length is found out from q_{int} (initial point) to the q_{goal} (goal point). For doing this all the paths from are found out, from which the path of minimum length is selected. The procedure for doing this is as follows: -

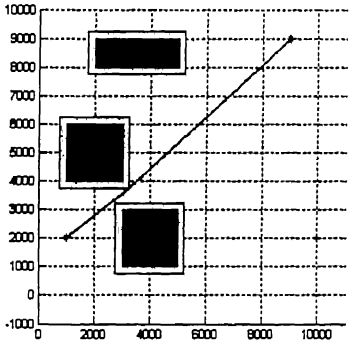
a) One node (n1) of any obstacle is searched which is directly connected to q_{int} by a link of the network (roadmap) and whose distance from q_{goal} is less than the distance between q_{int} and q_{goal}.
b) A node of any obstacle is searched which is directly connected to n1 by a single link of network.

c) The procedure of searching the node is continued until the searched node happens to be q_{goal}.

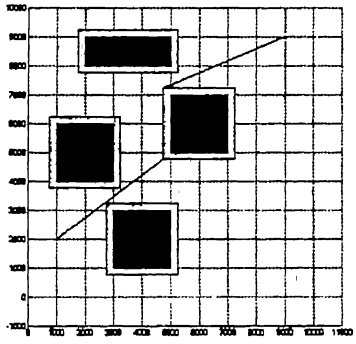
d) Steps a - c are repeated until all the paths have been searched.

e) The length of all the paths calculated and the path of minimum length is selected.

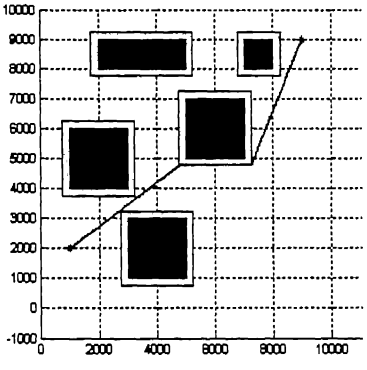
The path searched is in the form of nodes only and the whole path is generated by connecting the consecutive nodes by a line and taking finite number of points (e.g. 20) on the line, hence the total number of points on the path will be 20*(n-1), where n is the number of obstacles. Figure 5.16 shows a few examples of how a path is searched.



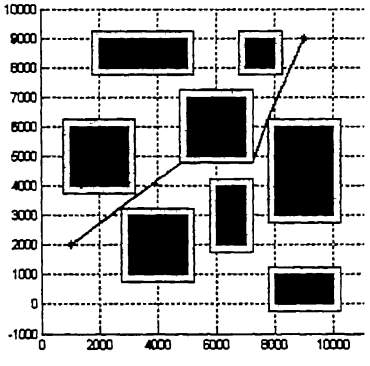
a) 3 obstacles



b) 4 obstacles



c) 5 obstacles



d) 8 obstacles

Figure 5.16. (a) – (d) Paths for different number of obstacles.

The path generated by the roadmap method is not smooth and it also touches the obstacle. Therefore to increase the distance of the path from the obstacle and to smooth the path potential method is used. In this approach artificial repulsive potentials are given by all the obstacles which push the path away from them.

The artificial repulsive potential produced by the obstacle must be such that:

a) It decreases with increase in the distance of the point from the obstacle, i.e. the force must be inversely proportional to the distance.
b) It must be very high for small distances, i.e. the force and distance relation should not be linear.
c) It must be repulsive, i.e. the direction of the force should be away from the obstacle.
d) It must be zero at end points.

By considering all these points, the following potential force is used:

$$F = -A/r^2 \times d1 \times d2 \qquad \text{------- (5.9)}$$

Here d1 and d2 are the distances of any point on the path from starting and end point respectively and r is the min distance of the point from the obstacle. After calculating the forces on each point, the points are moved in the direction of the force acting on that point. How much it is to be moved is calculated by using gradient search method in which the movement is calculated that will give maximum reduction in the forces.

5.2.2 Simulation for path planning

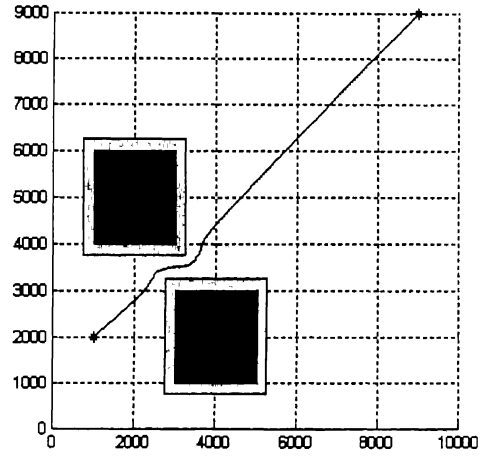
A MATLAB simulation has been developed for path planning among obstacles. The inputs to the simulation are as follows

a) Boundaries of the work space.
b) Number of obstacles in work space.
c) Nodes of the obstacles.
d) Starting point of the robot.
e) Goal point.

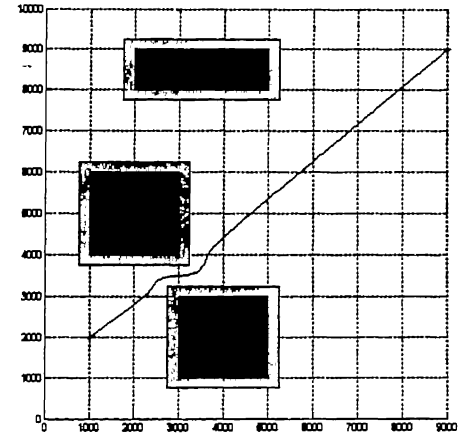
The output of the simulation is the set of points on the path between starting point and the goal point avoiding obstacles.

5.2.3 Simulation results of path planning

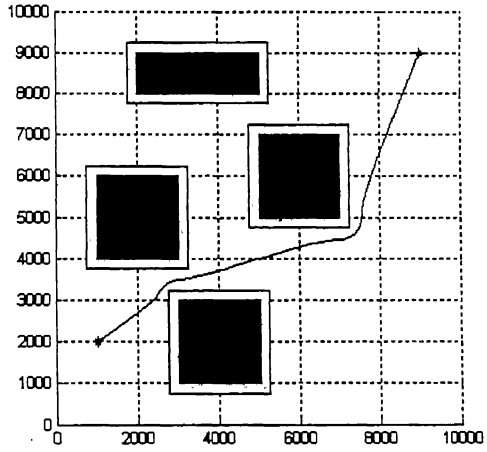
The results of the path planning are shown in Figure 5.17 (a-d) for 2, 3, 4 and 6 obstacles respectively.



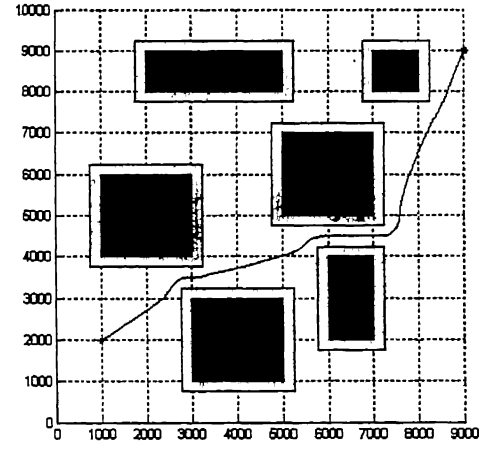
(a) path between 2 obstacles



(b) path between 3 obstacles



(c) Path between 4 obstacles



(d) path between 6obstacles

Figure 5.17. Results of path planning.

5.3 Stair climbing by the statically stable biped robot

The statically stable biped robot can not only walk and turn on flat ground but it can also climb up stairs, and move on slightly inclined ground. Since the robot is supposed to be balanced always, only those stairs in which the width and the depth of the step are such that the centre of gravity may be placed inside the step on which the leg is placed can be climbed up. As explained earlier while walking on a flat surface, one leg is raised up and moved forward and the other leg remains on the ground. After 180° rotation of the hip motors, the leg in air comes in contact with the ground and further rotation is resisted by the reaction forces from the ground. Next the second leg starts motion and comes forward. In climbing stairs, the leg in air is raised up and moves forward. At 90° rotation of hip motors, the leg reaches to the maximum height and after that it decreases. At height 'h' (height of stair) the leg contacts with next stair and further rotation of that leg is seized by reaction from the stair. The next leg now starts its motion. Figure 5.18 shows the step climbing by the robot:

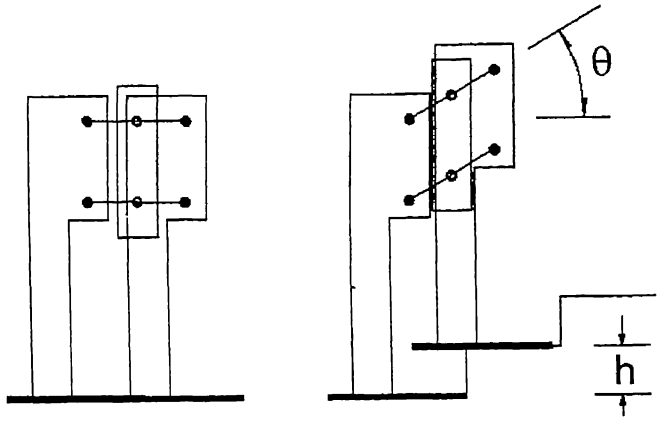


Figure 5.18. Stair climbing by the robot.

In straight walking on flat ground for one step the C.G. moves by 200mm, while the hip motors rotates by 180°. But in case of climbing stairs the angle of rotation of the hip would be proportional to the height of the step. When the leg in air touches the stair the four bar links attached to the body will have an angle satisfying the relation;

$$\mathbf{100 \times \sin(\theta) = h} \qquad \text{---------- (5.10)}$$

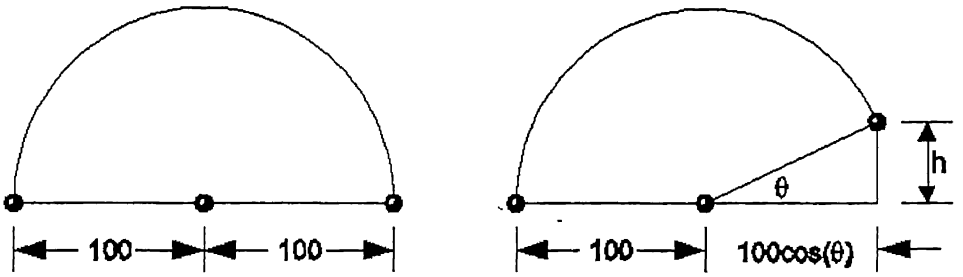


Figure 5.19. CG movement in walking on flat ground and climbing up stairs.

Based on the height of the stair the angle of the four bar link can be calculated as shown in Figure 5.19. The centre of gravity will travel by a distance equal to d, where

$$\mathbf{d} = 100 + 100 \times \cos(\theta) \qquad \text{---------- (5.11)}$$

The distance "d" will be less than 200 for stairs. Therefore the depth of the stair must be more than "d". Since there is an overlapping region in the foot perimeters, there will be an overlapping region in the depth of the stair as shown in the Figure 5.20 below.

Figure 5.20. Overlapping regions in the stairs.

Due to this overlapping portion step climbing for the statically stable biped robot becomes difficult. This is because when the leg in air touches the upper stair the second leg starts climbing up, but as soon as this leg reaches the upper stair it strikes the upper stair in this overlapping region. There are three different methods by which this problem can be solved.

a) Designing special stairs.
b) Rotating the foot in air.
c) Adding one more degree of freedom to bend the torso.

5.3.1 Stair climbing on specially designed stairs.

The stair can be designed such that it will have one slot cut length wise, which can allow the other foot to pass through. In this case only two degree of freedoms is necessary; the actuators of ankles will not be used for climbing the stairs. This would not be very practical as special stairs must be made for the robot and these stairs having cut would be unsuitable for humans. The design of the stairs and the placement of the feet on the stair are shown in Figure 5.21.

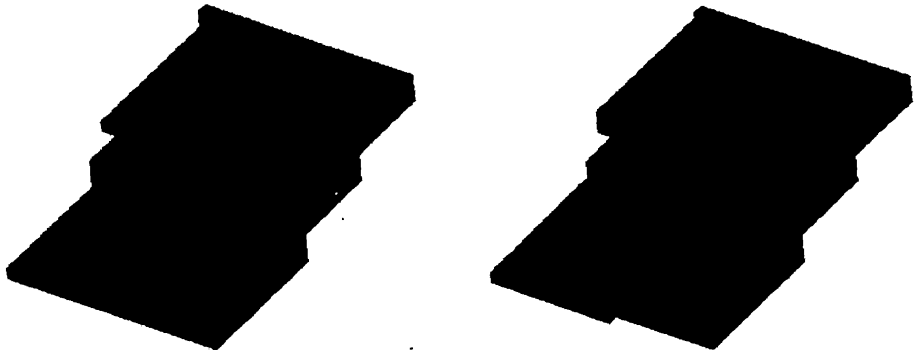


Figure 5.21. Special design stairs.

5.3.2 Stair climbing by rotating the foot in air

When one foot is on the stair and the other leg is on the lower step, then the lower foot can be rotated by 180° outwards as it moves up. This would ensure that the lower leg does not strike the stair. Similarly when the leg in air comes closer to the upper stair where it is supposed to stand, it can strike the next upper stair also, therefore again it is rotated by 180°. In this way stairs can be climbed by using a rather cumbersome way of walking.

5.3.3 Stair climbing by bending the trunk

One more degree of freedom can be added to the trunk, such that the torso can move back and forth. By doing this the C.G. travel can be reduced and the width of the stair needed also reduces. Therefore the stairs can be normal and the gait used for climbing also will be normal. This however requires an additional degree of freedom to

be added to the robot. This approach appears to be the most practical of the three mentioned above. Figure 5.22 shows stair climbing by bending the torso.

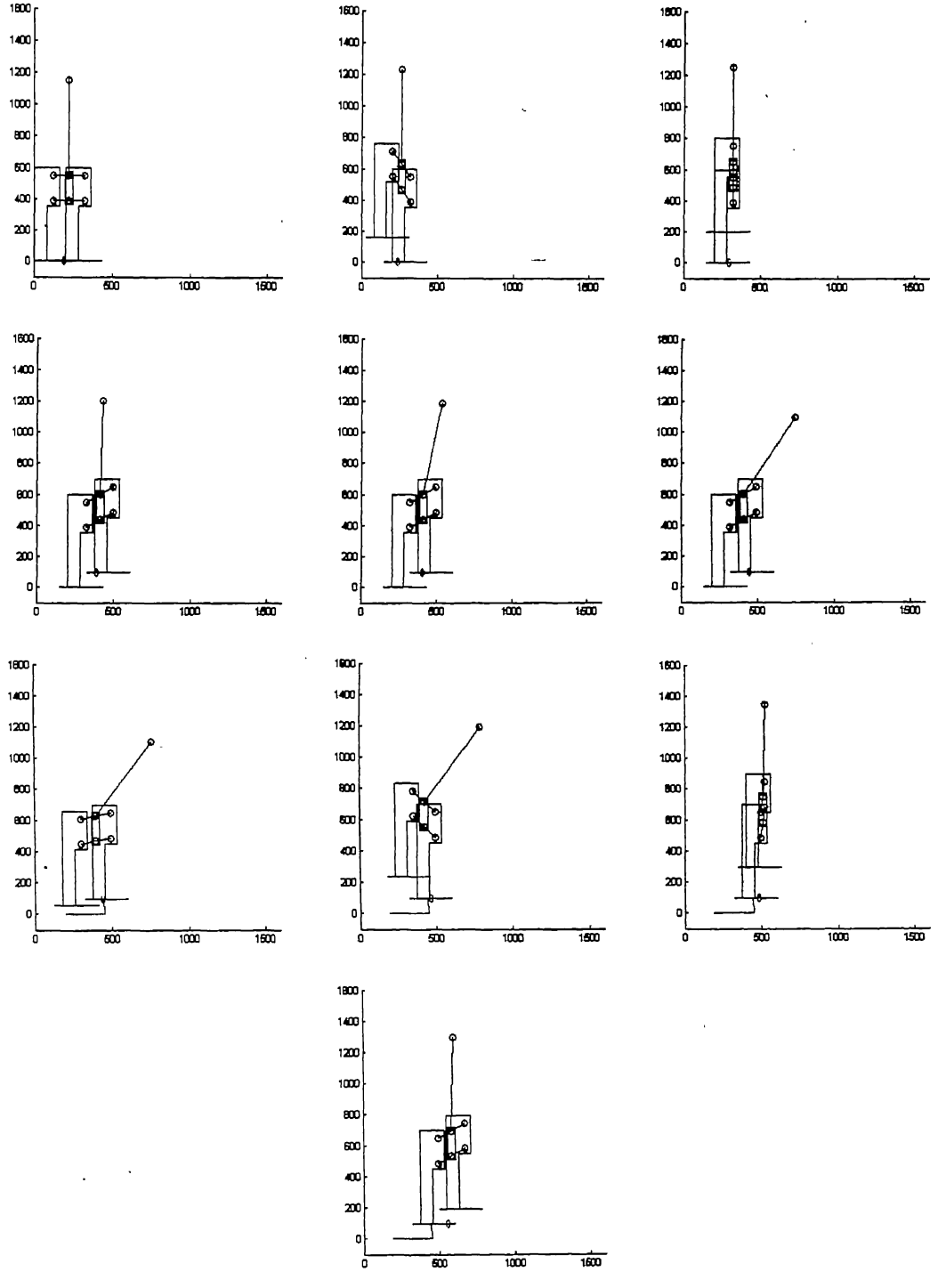


Figure 5.22. Results of stair climbing by bending trunk.

Chapter 6

Experiment and Results

The developed robot was fabricated in a local industry and it is shown in Figure 6.1. Experiments were carried out in two steps. In the first step the robot was made to follow a straight line. A simple program was written in C-language which gave the required angle of rotation of the motors at the hip-joints to the PMAC controller. In the program keeping the ankle motors constant the robot was made to take three steps. The top of the robot was marked by a florescent circle and using an overhead vision system the position of the robot was captured as it moved. The motion of the robot for one step is also shown by a series of photographs in Figure 6.2. Using the vision system the CG of the circle (robot) was located and compared with the desired straight line. The result is plotted in Figure 6.3. The result proves that the robot actually follows a straight line. The errors in the position of the CG of the robot with respect to the straight line are caused by the flexibility in the legs and ankle of the robot.

In the second experiment the robot was made to follow a small curve as in the previous experiment the centre of the robot was obtained by an overhead vision camera. The Figure 6.4 shows the result of robot on the curved trajectories. The result proves that the statically stable biped robot can follow simple curves. The errors here are also caused by flexibility and they tend to increase as the robot is repeatedly turning only on one side. Most trajectories can be decomposed into straight line segments and curves. Hence the robot can follow most trajectories.

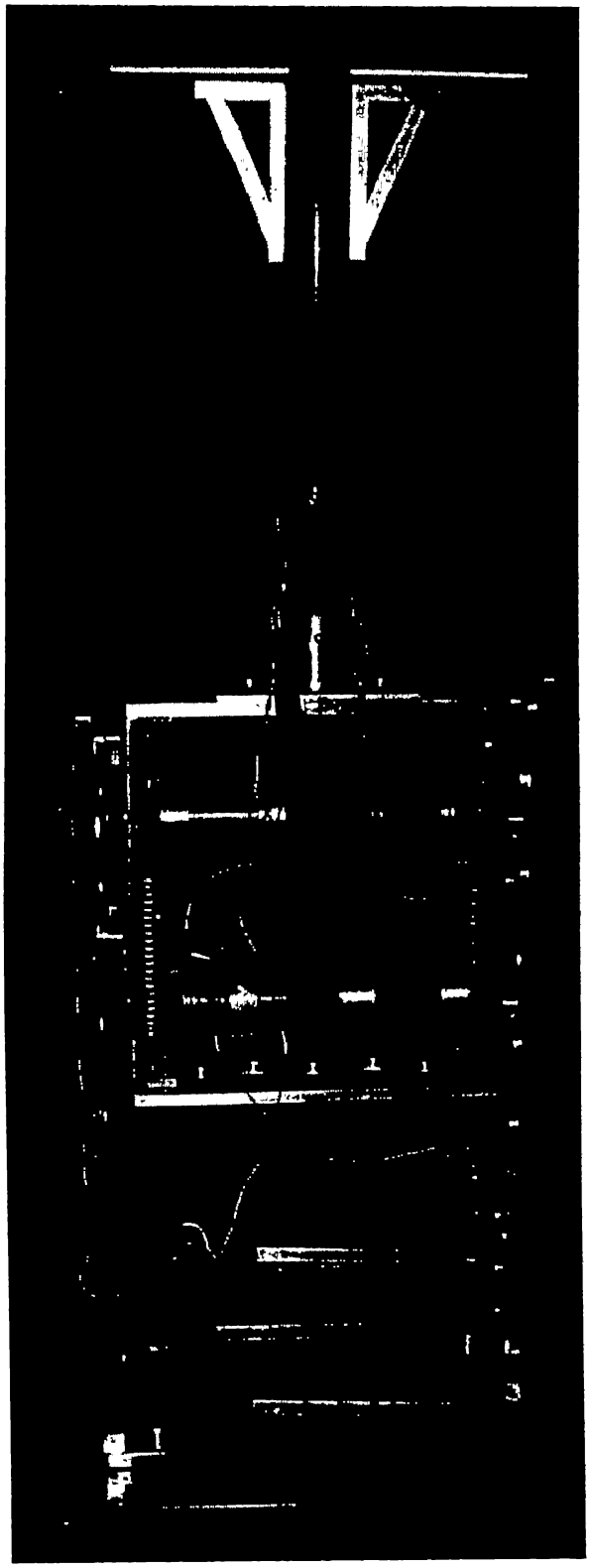


Figure 6.1. Statically stable biped robot.

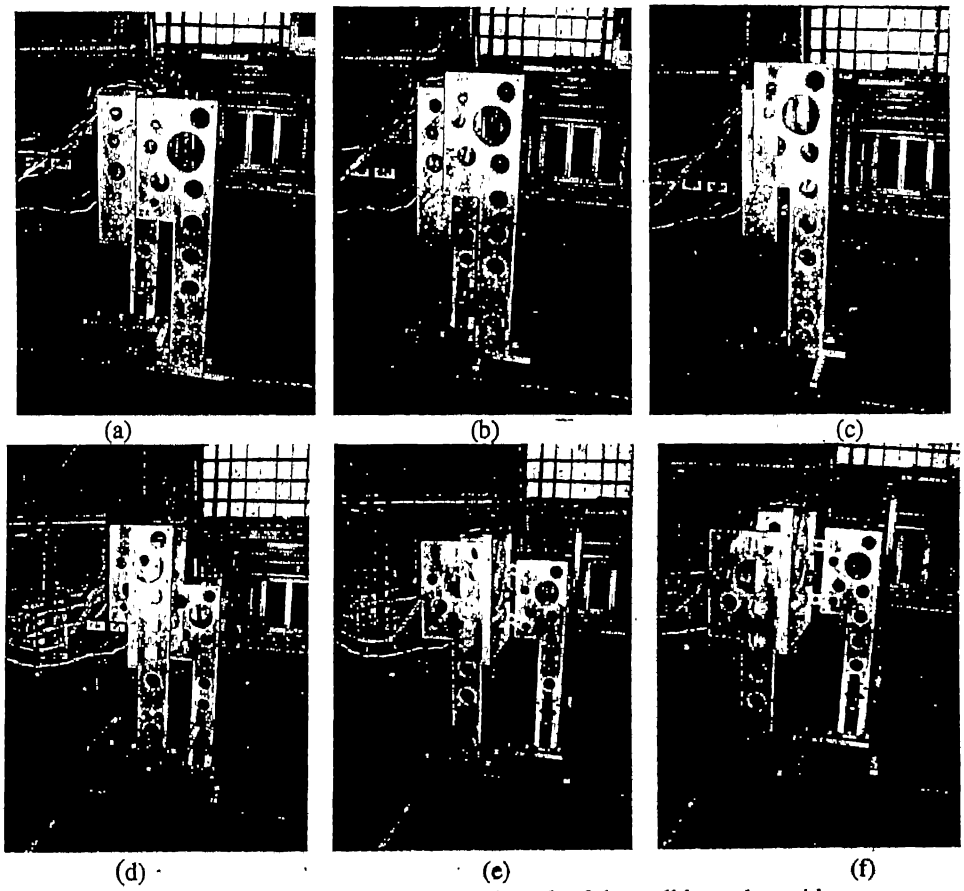


Figure 6.2. (a) – (f) Shows the experimental result of the walking robot taking a step.

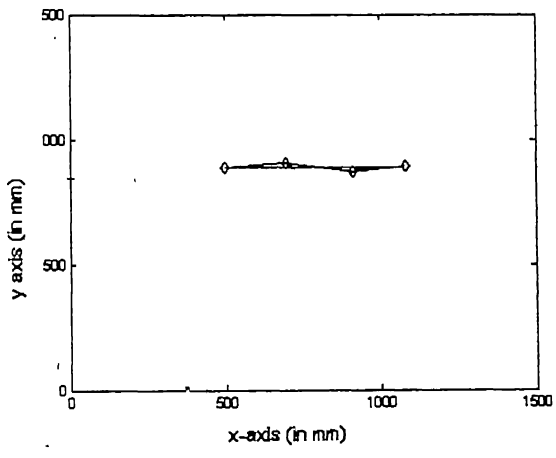


Figure 6.3. Comparison of actual and desired CG path for walking on a straight line.

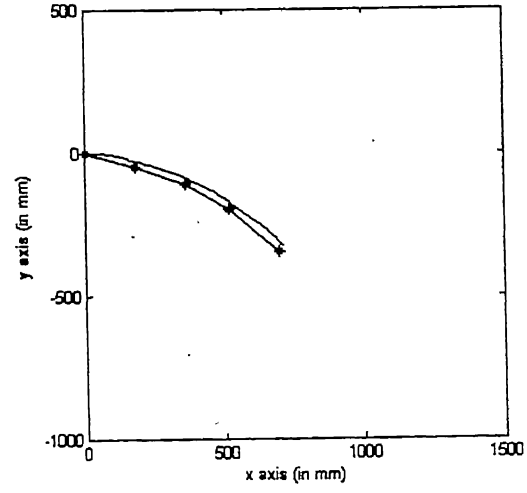


Figure 6.4 Comparison of the actual and desired CG path in case of turning.

Chapter 7
Conclusion

In this thesis, first a detailed study of the different types of biped (dynamic and static) that have been developed by researchers around the world was carried out. It was observed that all dynamic bipeds developed so far have human leg like structure. This requires the control of many actuators (10 or more) which makes it very complicated. In order to reduce the complexity, a simple statically stable biped robot was designed, developed and experimentally evaluated.

The mathematical analysis of biped gait based on ZMP control provided an insight into the dynamics involved in gait and its control. A simulation for dynamic walk was made in MATLAB which proved that dynamic stability during biped gait can be obtained by controlling the ZMP. A detailed study of motor torque required during gait was also carried out. It was found that the maximum torque is required by the ankle motors. The least torque is required by the hip motor, as it only has to move the leg in air forward. These results prove very useful in designing a dynamic biped in the next phase of the project.

A statically stable robot was designed, developed and experimentally evaluated. A detailed study of different mechanisms showed that the four bar mechanism was suitable for lifting the leg and moving it forward during gait. The dimensions of the foot were such that the robot was stable even on one leg stance. A simulation of the robot was made in MATLAB. The results of the simulation prove that the robot can follow a trajectory specified by given control points. For following a trajectory, minimum radius of curvature of trajectory was found to be 1.3 m without back stepping; while with a single back stepping it can follow a curve of 0.5 m radius. It can follow curves of radius less than 0.5 m using multiple back steps. The robot can also climb stairs by moving its torso to change the position of the CG. Two experiments were carried out in which the biped robot was made to walk in a straight line and also to follow a curve. The results prove that the robot can actually walk straight and also follow a curve with only four actuators.

The developed statically stable biped robot is the simplest biped robot that can follow trajectories. The main advantage of the robot is that its control is simple, and its cost as compared to the other humanoid robots would be very much cheaper. However as its structure is not similar to the human leg structure, its gait will not be similar to human gait. Also as it does not have redundant degrees of freedom, it would not be able to actively charge its CG to balance against severe external disturbances, like a strong push or pull.

There is a lot of scope for further research based on the work presented in this thesis. The ZMP control simulation has proved that a biped can be controlled by changing the ZMP. However, a biped should be fabricated and then the ZMP based control scheme experimentally evaluated. Two hands should be put on the biped so that it could do useful work. Also a study should be undertaken to find out the minimum number of actuators required for stable work (as in underactuated systems). There is a lot of scope for adding other features. Speech recognition could be integrated with the controller so that the robot could respond to speech commands. Also, a vision system could be added for autonomous navigation.

Bibliography

[1] T. McGeer, "Passive Dynamic Walking", International Journal of Robotics Research, 9(2):62-82, 1990.

[2] Mariano, Chatterjee and Andy Ruina, "Efficiency, speed, and scaling of 2 D passive dynamic walking," Dynamics and stability of systems, July 1998.

[3] Coleman At el, "Passive Dynamic models of human gaits".

[4] Collins at el. "A Three-Dimensional Passive-Dynamic Walking Robot with Two Legs and Knees," International Journal of Robotic Research Vol. 20, No. 7, July 2001, pp. 607-615.

[5] Coleman et al. "stability and Chaos In Passive Dynamic Locomotion"

[6] Coleman at el. "The Simplest Walking Model: Stability, Complexity, and Scaling," ASME Journal of Biomechanical Engineering, Vol. 120 No. 2 pp. 281 - 288, April 1998.

[7] Coleman, Ruina "An Uncontrolled Toy That Can Walk But Cannot Stand Still," A, Physical Review Letters April 1998, Vol. 80, Issue 16 pp. 3658 - 3661.

[8] John Camp, "Powered "Passive" Dynamic Walking" Masters of Engineering Project Report, Cornell University, Ithaca NY.

[9] Sugimoto, Osuka, "Walking Control of Quasi-Passive-Dynamic-Walking Robot "Quartet III" based on Delayed Feedback Control".

[10] Raibert, Brown, "Experiments in balance with a 2D one-legged hopping machine," ASME Journal of Dynamic Systems, Measurement and Control 106, pp. 75-81.

[11] Hodgins, Koechling, Raibert, "Running experiments with a planner biped," Third International Symposium on Robotics Research, Cambridge: MIT Press.

[12] M Y Zarrugh and CW Radcliffe, "Computer generation of human gait kinematics", Journal of Biomechanics, vol.12, pp. 99–111, 1979.

[13] P. H. Channon, S. H. Hopkins, and D. T. Phan, "Derivation of optimal walking motions for a biped walking robot," Robotica, vol. 10, no. 2, pp. 165–172, 1992.

[14] M. Rostami and G. Bessonnet, "Impactless sagittal gait of a biped robot during the single support phase," in Proc. IEEE Int. Conf. Robotics and Automation, 1998, pp. 1385–1391.

[15] L. Roussel, C. Canudas-de-Wit, and A. Goswami, "Generation of energy optimal complete gait cycles for biped robots," in Proc. IEEE Int. Conference of Robotics and Automation, 1998, pp. 2036–2041.

[16] F. M. Silva and J. A. T. Machado, "Energy analysis during biped walking," in Proc. IEEE Int. Conf. Robotics and Automation, 1999, pp. 59–64.

[17] Y. F. Zheng and J. Shen, "Gait synthesis for the SD-2 biped robot to climb sloping surface," IEEE Trans. Robotics and Automation, vol. 6, pp. 86–96, Feb. 1990.

[18] C. Chevallereau, A. Formal'sky, and B. Perrin, "Low energy cost reference trajectories for a biped robot," in Proc. IEEE Int. Conf. Robotics and Automation, 1998, pp. 1398–1404.

[19] Vukobratovic, Frank and Juricic, "On the Stability of Biped Locomotion," IEEE Transaction on Biomedical Engineering, BME-17, No. 1, pp. 25-26, 1970.

[20] A. Takanishi, M. Ishida, Y. Yamazaki, and I. Kato, "The realization of dynamic walking robot WL-10RD," in Proceeding of International Conference on Advanced Robotics, 1985, pp. 459–466.

[21] C. L. Shih, Y. Z. Li, S. Churng, T. T. Lee, and W. A. Cruver, "Trajectory synthesis and physical admissibility for a biped robot during the single support phase," in Proceeding of IEEE International Conference on Robotics and Automation, 1990, pp. 1646–1652.

[22] C Shih, "Gait synthesis for a biped robot," Robotica, vol. 15, pp. 599–607, 1997.

[23] C Shih, "Ascending and descending stairs for a biped robot," IEEE Transaction on System, Man.,.Cybern. A, vol. 29, no. 3, 1999.

[24] K. Hirai, M Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in Proceeding of IEEE International Conference on Robotics and Automation, 1998, pp. 1321–1326.

[25] A Dasgupta and Y. Nakamura, "Making feasible walking motion of humanoid robots from human motion capture data," in Proceeding of IEEE International Conference on Robotics and Automation, 1999, pp. 1044–1049.

[26] Qiang Huang "Planning Walking Patterns for a Biped Robot", IEEE Transaction on robotics and automation, vol. 17, no. 3, June 2001.

[27] Ching-Long Shih, "Inverted Pendulum-Like Walking Pattern of a 5-Link Biped Robot," ICAR '97 Proceedings of 8th International Conference on Advanced Robotics, 7-9 July 1997 Page(s): 83 -88.

[28] Kajita, S., Tani, K. "Experimental study of biped dynamic walking in the linear inverted pendulum mode," Proceedings on IEEE International Conference on Robotics and Automation, 1995, Volume: 3, 21-27 May 1995 Page(s): 2885 -2891.

[29] Caux, Mateo, Zapata, "Balance of Biped Robots: Special double inverted pendulum," IEEE International Conference on Systems, Man, and Cybernetics, 1998, Volume: 4, 11-14 Oct. 1998 Page(s): 3691 -3696.

[30] Park and Kim, "Biped Robot Walking Using Gravity-Compensated Inverted Mode and Computed Torque Control," Proceedings of the IEEE International Conference on Robotics & Automation, May 1998, page no. -3528-3533.

[31] Sughihara, Nakamura, Inoue, "Real-time Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control" Proceedings of the 2002 IEEE International Conference on Robotics & Automation May 2002, page no.-1404-1409

[32] Tsuji, T.; Ohnishi, K., "A control of biped robot which applies inverted pendulum mode with virtual supporting point," 7th International Workshop on Advanced Motion Control, 2002, Page(s): 478 -483

[33] Kurazume, R., Hasegawa, T., Kan Yoneda, "The sway compensation trajectory for a biped robot," Proceedings of IEEE International Conference on Robotics and Automation, ICRA '03, Volume: 1, Sept. 14-19, 2003, Page(s): 925 -931.

[34] Hashimoto S, Narita S., Kasahara H. et al. "Humanoid Robots in Waseda University – Hadaly –2 and WAIBIAN," Autonomous Robots. 12 (2002) 25-38

[35] Hirai K., Hirose, M., Haikawa. Y., Takenaka, T. "The development of the Honda Humanoid Robot," Proc. of the IEEE Intl. Conf. On Robotics and Auto. (1998) 1321-1326.

[36] Ching-Long Shih and Chien-Jung Chiou, "The Motion Control of a Statically Stable Biped Robot on an Uneven Floor", IEEE Transactions on systems man and cybernetics-part B: CYBERNETICS, vol. 28, no. 2, APRIL 1998.

[37] Kulkarni, Dutta, abhishek, mukarjee, "Design of an intelligent statically stable Humanoid robot," Proceedings of the National Conference on Advanced Manufacturing and Robotics, India, 2004 (In Press).